COMPUTER TEACHING-AIDS FOR AN UNDERGRADUATE COURSE IN DISTILLATION

BY

JOHN MALCOLM TRAVERS GREENSMITH

Submitted to the University of Cape Town in fulfilment of the requirements for the degree of Master of Science in Engineering

April 1976

ACKNOWLEDGEMENTS

The author wishes to express his sincere thanks and appreciation to:

Mr. B.D.A. Paddon for his guidance during the course of this study,

Messrs. D.E. Stegmann and L.D.A. Wright of the Computer Centre, but for whom using the UNIVAC would be somewhat more difficult, and

Ms. P. Coetzee for her time and patience in typing this thesis.

Finally, the financial assistance of the Council for Scientific and Industrial Research, and the Department of Chemical Engineering, U.C.T., is gratefully acknowledged.

ABSTRACT

Four computer programs dealing with the following aspects of distillation are developed as an integrated teaching package.

(1) Binary batch distillation in tray columns.
(2) Binary continuous distillation in tray columns.
(3) Binary continuous distillation in packed columns.
(4) Multicomponent distillation in tray columns.

A plotting program utilizing CALCOMP plotting software and hardware is developed for graphical representation of the results generated by the above routines. The following diagrams are included:

(1) McCabe-Thiele x,y diagrams for batch and continuous distillation.
(2) Ponchon-Savarit diagram for continuous distillation.
(3) Temperature, flowrate, and composition profiles for multicomponent distillation.

The programs are designed to supplement an undergraduate course in distillation at U.C.T. Mathematical and computational techniques used in the programs correspond to material presented in the course. A mainline-subroutine program structure is used throughout to facilitate future additions and alterations to the program.

The routines compare favourably with similar published teaching programs, with respect both to computer-time and to core-storage requirements. Program solutions to tutorial problems agree well with manual textbook calculations, in the design of tray columns. The program for designing packed columns has revealed discrepancies in the application of certain mass transfer correlations to distillation systems.

The CACHE format for describing programs is adopted in this text.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION AND LITERATURE SURVEY

1.1 INTRODUCTION

This study is concerned with the development of computer programs to be used as teaching aids in an undergraduate course on Distillation at the University of Cape Town.

1.11 Historical Background

At the turn of the century distillation was more an art than a science, being associated mainly with the French wine industry. The basis of modern distillation theory was presented by Ernest Sorel (1893) in his book "La Rectification de l'alcool" and the first English text on the subject appeared in 1903. There was little progress in distillation technology, however, until the 1920's, when the distillation of crude oil attracted the interest of a number of research workers. Two algebraic techniques for solving steady-state multicomponent distillation problems were developed by Lewis and Matheson [1], and Thiele and Geddes [2]. These procedures form the basis of most modern computer algorithms. Robinson (1930) published the first teaching text on distillation, which was later to be co-edited by Gilliland [3].

Unfortunately the manual calculation methods of Lewis and Thiele proved to be tedious and time-consuming. Simplifying assumptions, that were necessary in most cases, reduced the precision of the techniques and engineers tended to rely on experience and rule-of-thumb design methods. In 1950, when the fourth edition of Robinson's book was published, Gilliland noted that for a number of years the "law of diminishing returns" had been applying to distillation research.

The late 1950's saw the advent of large, high-speed digital computers and interest in rigorous design methods was revived. The first computer algorithm for solving distillation problems was published by Amundson and Pontinen [4] in 1958. Several more followed soon afterwards, and throughout the 1960's considerable research effort was directed towards either improving the stability or reducing the computer-time requirements of the various procedures.

More recently it has been recognised that the accuracy of equilibrium, enthalpy and stage efficiency data is often the limiting factor in design or simulation of distillation systems. Ellis and Boyes [5] have recently reviewed the present extent of knowledge in distillation, indicating fields in which research is most urgently required.

Distillation is now a vital part of many industrial processes. Modern computers make possible the use of mathematical models that were previously too complex for convenient use. Zuiderweg [6] has estimated that research in distillation has saved industry a possible $2000 million over the past 20 years.

Despite these advances the content of texts dealing with the subject at the undergraduate level has not changed appreciably since 1950. This study is concerned with the needs of students rather than qualified engineers.

1.12 Objectives

The chemical engineer is expected to be familiar with all the separation processes commonly used in industry. Consequently undergraduate courses on these topics are invariably congested, and there is little time for more than a general treatment of each process. At U.C.T. a third-year course in mass-transfer operations presently allows approximately nine lectures and three tutorial sessions for distillation processes.

In this time students are taught the basic principles of both binary and multicomponent distillation, and must apply these principles to textbook problems. As even elementary problems can be time-consuming, solution procedures are restricted to the graphical techniques of McCabe and Thiele [7] and Ponchon and Savarit [8] for binary systems, and to various short-cut methods for multicomponent problems.

Due to the limited time available, problems are usually solved for a single fixed set of conditions. Although the effects of reflux ratio, feed condition, stage efficiency and other variables are discussed theoretically, they are seldom demonstrated in tutorials. As a result, the average student's understanding of distillation is often inadequate.

It was proposed that a set of computer programs be developed, capable of solving the problems normally set in tutorials. The programs would be made available to students who would then be able to:

1) test manual solutions,
2) investigate the effect of changes in key variables, and
3) analyse the results and comment on the operating conditions specified in a problem.

The objective of this study was to develop such a set of programs for solving distillation problems. The following requirements and programming guidelines were specified:

1. Type of distillation problem considered:
   (a) Binary Batch distillation in Tray towers
   (b) Binary Continuous distillation in Tray towers
   (c) Binary Continuous distillation in Packed towers
   (d) Multicomponent Continuous distillation in Tray towers.

2. Structure of Programs:
   Modular, mainline-subroutine structure to be adopted, with generalised subroutines for standard

calculations such as bubble point or dew point. Programs to be structured to minimise core storage requirements.

3. Operation of Program:

   Both Demand- and Batch-mode operating facilities to be provided, with emphasis on user-program interaction in Demand-mode.

4. Computer Algorithms:

   Mathematical correlations and solution procedures to correspond to those discussed in the course lectures.

5. Data Collation:

   Optional data input facilities to be included to minimise the time required to set up problem data tables.

6. Error messages:

   Common solution errors, such as pinch points, to be searched for, and remedies suggested through program error messages.

7. Graphical solution procedures:

   A plotting program to be developed for construction of McCabe-Thiele or Ponchon-Savarit diagrams for binary systems, and temperature, composition and flow profiles for multicomponent systems.

Further requirements were:

(a) Thorough testing of programs using textbook examples;

(b) A comprehensive User Manual for the program package.

1.2 LITERATURE SURVEY

This survey is divided into four sections. The first deals with availability and utilization of computer programs at the undergraduate level. Section 1.22 reviews distillation processes with specific reference to the undergraduate-course content at U.C.T. The selection and adaptation of solution procedures for the various programs are discussed in section 1.23. The last part is a detailed review of the numerous computer algorithms that are available for multi-component distillation design.

1.21 Use of computers for teaching purposes

Although computers have been used in industry for many years, for both design and control of equipment, their application in undergraduate courses has been fairly limited. In the past the only contact many students had with a computer was a basic course in FORTRAN programming, often given by a department outside the Engineering faculty. Wellknown texts on distillation, and other separation processes, such as Treybal [9] and McCabe and Smith [10] mention computerised solution procedures briefly, but do not consider them in tutorial examples. Buford Smith [11] gives a concise description of four computer algorithms for solving multi-component problems, but states that his book is written primarily for practising engineers.

Recent cost escalations have created a demand for more efficient design and control of industrial equipment. Computers are being used more extensively and engineers consequently require a better understanding of their application and capabilities. Since 1970 several universities in the U.S.A. have included computer programming as an integral part of courses in their Chemical Engineering curricula, rather than as a separate subject. The CACHE* committee has compiled a comprehensive set of programs covering a wide range

* CACHE - Computer Aids for Chemical Engineering Education.

of subjects. Many of the contributors are Chemical Engineering lecturers. All the programs are written specifically for undergraduate use and are readily available at a reasonable fee, in contrast to commercial programs which are often extremely expensive.

A recent text on distillation by Thibault-Brian [12] of M.I.T. deals extensively with computerised solution procedures. In common with the approach of several CACHE contributors, the tutorial examples require students to write their own programs to solve problems. With the limited computer facilities available at U.C.T., and the short duration of the course, it would not be practical to adopt a similar system here.

On examining the CACHE programs on distillation [13] it was found that they were, in general, unsuitable for the type of facility required at U.C.T. Many of them were merely examples of the type of program required of students as a tutorial solution. As all the programs were contributed by different authors they did not form an integrated package. Solution procedures used in some of the programs did not correspond to the U.C.T. course-content. Enthalpy effects were not considered at all in programs dealing with binary systems. A multicomponent algorithm required equilibrium and enthalpy data to be supplied as polynomials in temperature. Data for the sample problem shown with this program had been extracted from Amundson's paper [4]. To supply data in this form for new examples is usually inconvenient.

It was decided that a unique set of programs should be developed as an integrated package. The programs would be designed to supplement a course on distillation given at U.C.T.

1.22 Distillation processes

Separation processes have as their common aim the isolation of a desired component, or group of components, from a mixture of the desired product and impurities. Distillation

is the separation, by vaporization, of a liquid mixture of miscible, volatile substances. The degree of separation achieved depends upon the distribution of the components between the two phases.

Specifically, the word "distillation" applies to a single vaporization process in which a liquid mixture is heated and the evolved vapour recovered, usually by condensation. If the vapour is kept in continuous, counter-current contact with a condensate, the process is termed "rectification". Usually, however, the two words are considered synonymous.

The principle of separation by successive vapour-liquid phase-distribution operations gives rise to the concept of an "equilibrium stage". The basic assumption of an equilibrium stage is that contact between the initial vapour and liquid phases within the stage is sufficient for the system to reach thermodynamic equilibrium. This concept is used to design a system of contact devices that will achieve the required degree of separation.

Counter-current contacting of the vapour and liquid phases is usually achieved in a column containing either trays or a packing material. In tray columns each theoretical tray represents an equilibrium stage whereas in packed column the stage is represented by a fixed height of packing. This height equivalent to a theoretical plate (HETP) is partly dependent on the characteristics of the packing material. Theoretical trays, or plates, are trays on which a true equilibrium is achieved. The performance of a real tray is related to an ideal equilibrium stage by a proportionality factor termed a stage-efficiency.

Modelling real trays in terms of ideal trays and efficiency factors is a far easier task than the direct solution of the complex heat and mass transfer operations associated

with a real tray. All solution procedures considered at the undergraduate level are based on ideal models that include efficiency factors.

In industry, distillation processes are run as either batch or continuous operations. Either operation can be carried out in tray or packed columns, to separate either binary or multicomponent solutions. The incidence of true binary solutions in practice is small, however, and consequently most of the research in distillation has involved multicomponent systems. The processes considered here have been listed in section 1.12.

Traditionally, in an introductory course on distillation binary systems are discussed first. The multicomponent process is introduced later as a complex extension of the binary case. The four basic equations describing the generalized equilibrium stage (Figure 1.1) may be expressed as follows:

1. The equilibrium relationship on stage j

$$y_{i,} = K_{i,j}\, x_{i,j} \qquad (1.1)$$

2. The component material balance around stage j

$$l_{i,j} + v_{i,j} - l_{i,j+1} - v_{i,j-1} - f_{i,j} = 0 \qquad (1.2)$$

3. The enthalpy balance around stage j

$$L_j h_j + V_j H_j - L_{j+1} h_{j+1} - V_{j-1} H_{j-1} - F_j h_{fj} + q'_j = 0 \qquad (1.3)$$

4. The restriction on fractional concentrations on stage j

$$\Sigma x_{i,j} = 1{,}0 \quad \text{and} \quad \Sigma y_{i,j} = 1{,}0 \qquad (1.4)$$

The terms in the above equations are defined in the nomenclature. (Appendix B)

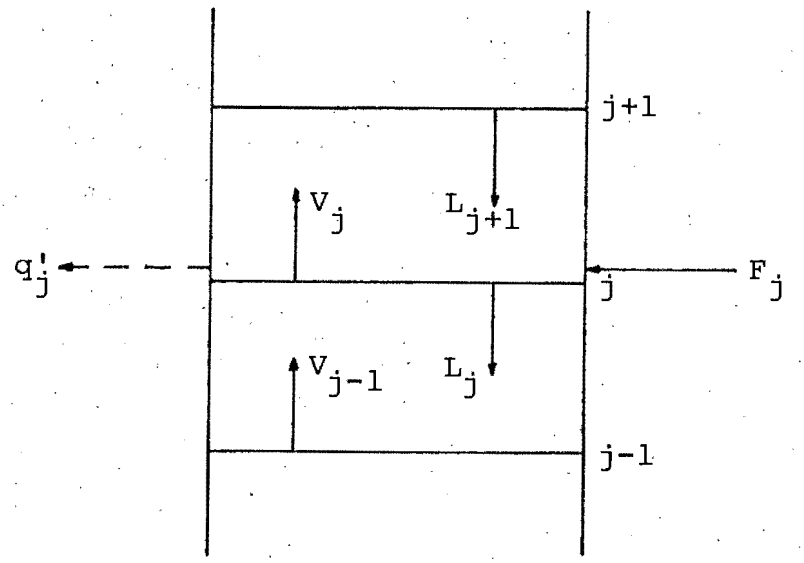


Fig. 1.1

Generalised equilibrium stage

Friday and Smith [14] note that there are many ways of combining and solving these equations. This has resulted in the development of a large number of solution procedures for multicomponent systems. For binary systems the solution of the above equations can be conveniently represented graphically in two dimensions.

1.23 Solution Procedures

The calculations required to solve any distillation problem can be divided into three parts.

1. Prediction of phase equilibria.
2. (a) Calculation of the number of equilibrium stages required for a given product distribution, OR
   (b) Calculation of the separation achieved in a column containing a specified number of equilibrium stages.
3. Conversion of the number of equilibrium stages to real stages.

The second part of the solution procedure is the simplest and is the only aspect discussed in any detail in the undergraduate course. The first and third parts are extremely complex, usually involving empirical or simplified mathematical models. It is in these fields that further research is required [5].

A. Binary distillation in Tray towers.

All distillation problems considered in the course involve ideal mixtures that obey the equilibrium expressions shown previously in Equation (1.1). The equilibrium constant K_i for binary systems is derived from Raoult's Law and Antoine's equations (see Chapter 2). The graphical solution procedures of McCabe and Thiele [7] and Ponchon and Savarit [8] are considered in this part of the course.

Batch distillation is introduced first and standard McCabe-Thiele assumptions are applied to all systems considered. The problems are thus solved manually by constructing McCabe-Thiele diagrams, with the necessary integration performed graphically as well. These procedures are well documented in a number of texts [9,10,11] and a review of the literature did not reveal any new solution procedures. A recent paper by Koppel [15] describes a computerised McCabe-Thiele solution procedure, but does not suggest any significant changes in method. Batch distillation systems operating under both constant and variable reflux ratio are considered. Chapter 2 describes the program devised for binary batch distillation. The approach is similar to that of Koppel except that a Simpson's Rule numerical integration technique is used, instead of a graphical one. Stage-efficiencies are not considered in batch distillation.

Continuous binary distillation is considered next in the course. McCabe-Thiele systems are discussed first, with the McCabe-Thiele diagram used to demonstrate the overall effect of feed condition, reflux ratio, and other variables.

Enthalpy effects are then considered and the Ponchon-Savarit diagram is described.

Very little new work has been done in binary distillation. Strangio and Treybal [16] have developed a series of exact relationships, in algebraic and graphical form, that may be used for "short-cut" design of both binary and multi-component systems. The applications of these methods are not relevant to the course, however. The programs published by the CACHE committee ignore enthalpy effects and columns with side streams or more than one feed. They were thus unsuitable for this section of the course.

Stage-efficiency is discussed in the course in terms of Murphree tray efficiencies. The effect of efficiency is considered in the program, and an efficiency factor may be specified. The program contains both the McCabe-Thiele and the Ponchon-Savarit procedures and is discussed in Chapter 3. The output from either the batch or the continuous distillation programs may be used directly by the plotting program to construct McCabe-Thiele or Ponchon-Savarit diagrams. The plotting program is discussed in Chapter 6. Reference to plotting programs of this type could not be found in the literature.

B. Binary distillation in Packed towers.

Packed columns for distillation operations are usually designed manually, using empirical Height of Transfer Unit (HTU) or Height-Equivalent-to-Theoretical Plate (HETP) relationships. Empirical data are available for many different types of packing and the various design equations have been well tested over the years.

The concepts of HTU assumes that the vapour and liquid flowrates within each section of a column are constant. In a recent paper Eckert [17] states that vapour-liquid equilibrium is the only factor significantly affecting a binary distillation process, and thus justifies continued use of HTU relation-

ship to design packed columns. The more rigorous procedure, which includes enthalpy considerations and hence does not assume constant molal overflow conditions, requires the calculation of mass transfer coefficients at intervals through the column. The height of packing required is then determined by integrating the following expression between appropriate limits.

$$Z = \int_{Vy_1}^{Vy_2} \frac{d(Vy)}{k_y\, a(y_i - y)} = \int_{Lx_1}^{Lx_2} \frac{d(Lx)}{k_x\, a(x - x_i)} \qquad (1.5)$$

In manual calculations the integration is performed graphically and the accuracy depends on the integration interval considered. The procedure is tedious and extremely time-consuming with the result that any improvement in accuracy over an HTU method is not justified, especially as it has become standard practice to add extra packing length, in an almost arbitrary fashion, as a safety factor.

In terms of a computer teaching-package, however, the rigorous design procedure is a useful means of demonstrating both the concepts of mass transfer, and the possible variations in design obtained using the different techniques. Students can design a packed column manually using an HTU relationship, and then compare their results with a computer solution.

The effect of packing type and size can be demonstrated more easily using a computer routine. The use of mass transfer coefficients also enables the effect of physical properties of a system to be illustrated and shows how mass transfer theory is applied to process operations.

CACHE have not published a program dealing with packed column design for distillation purposes. Their program for

packed absorption tower design uses a mass transfer correlation procedure similar to that used here (see Chapter 8).

The program developed for packed tower design is discussed in Chapter 4. A Ponchon-Savarit operating diagram is used by the program whereas manual calculation procedures using an HETP method require a McCabe-Thiele diagram.

C. Multicomponent Distillation

Continuous distillation of multicomponent solutions is the most common distillation process encountered in industry. Design methods are more complex beacuse the process cannot be represented graphically. There are a number of solution procedures available of which one was chosen to meet the requirements of this study. Friday [14] was the first to analyse the possible combination of the four equations stated previously, and to characterise the corresponding solution procedure. He proposed a series of six "decisions" that would lead to the correct choice for a given problem. Seppala [18] applied sixteen different solution procedures to nine problems and showed that no single technique was superior in all cases. This supported Friday's contention that the nature of the problem should determine the choice of solution procedure.

The theory and application of the trial and error methods of Lewis [1] and Thiele [2] are discussed in the course. Due to the time factor, however, short-cut procedures, such as that of van Winkle and Todd [19], are usually used for problem solving. The short-cut method of van Winkle and Todd was shown by Guerreri [20] to produce results that agree well with those calculated by the Thiele-Geddes procedure. An extensive literature survey was conducted to determine the most suitable solution procedure for the multicomponent program. The survey is summarised in section 1.24.

The combination of the Thiele-Geddes calculation procedure and Holland's [21] 'θ' convergence technique was finally chosen for the following reasons:

(1) Results agree well with short-cut method of van Winkle and Todd which will facilitate comparison of short-cut and computer solutions.
(2) The Thiele-Geddes-θ combination is computationally fast and stable.
(3) It is a tray-to-tray calculation procedure that does not require an understanding of matrix algebra in order to follow the programming procedure.
(4) The Thiele-Geddes method is discussed in the course and is recognised as being more stable than the Lewis-Matheson procedure.

Ideal equilibrium relationships are assumed to hold for all problems considered because the scope of the course does not include non-ideal solution thermodynamics. Stage-efficiency is considered in terms of component vaporization efficiency. This procedure was first modified for computer application by Holland [21]. There is no realistic method for determining component efficiency factors analytically, but fairly representative results have been obtained experimentally [22]. Miskin [23] demonstrated that vaporization efficiency factors were more consistent than the more widely used Murphree efficiency factors. Vaporization efficiency is useful for demonstrating the effect of inefficiency on a system even if true data are unavailable. As yet there is no satisfactory way of either measuring or calculating reliable efficiency data [22].

The multicomponent distillation program is discussed in Chapter 5.

1.24 Survey of Multicomponent design procedures

Computerised calculation procedures for the rigorous solution of distillation problems may be divided into four main categories as follows:

Class 1: Programs based on solving the four equations listed in section 1.22 simultaneously for the whole column using matrix algebra.

Class 2: Programs that consider the dynamic problem of the column settling down to steady-state using a relaxation method.

Class 3: Programs based on the Lewis-Matheson [1] procedure.

Class 4: Programs based on the Thiele-Geddes [2] procedure.

Class 3 and 4 programs are similar in that they are both tray-to-tray methods. They involve different initial assumptions, however, and traditional class 3 programs are design programs whereas class 4 are used for rating columns containing a fixed number of trays. This, however, is not an essential feature of either type.

Class 1 programs involve the simultaneous solution of a set of equations describing a distillation system. The equations are set up in a matrix of order n, where n is the number of trays in the column including the condenser and reboiler. The matrix is reduced to a tridiagonal form that can be solved using either matrix-inversion of Gauss-elimination techniques. Class 1, 3 and 4 programs all consider columns operating at steady-state conditions.

Class 2 programs involve an entirely different approach, as they consider the dynamic problem of column's achieving steady-state conditions from start-up. A relaxation method is used to achieve a solution.

Each class of program will now be discussed in detail. They are listed in order of publication of the original papers. The reasons for selecting the procedure used in this study are then summarised.

(a) Class 1 Programs

Amundson and Pontinen [4] were the first to investigate the use of large digital computers for distillation calculations. They observed that the basic equations describing an equilibrium stage process, listed in section 1.22, could be grouped as a set of simultaneous algebraic equations in composition, that describe a complete distillation column.

Their proposed calculation procedure involves the simultaneous solution of this set of equations, component by component, for a fixed temperature profile. Successive iterations correct the temperature distribution until a convergence is achieved. The procedure is strictly valid only for ideal systems. In such systems, the equilibrium constant is not a function of composition and thus the fixing of the temperature profile is sufficient to ensure that the overall material balance is not violated during the calculation. Most low molecular weight hydrocarbon systems may be considered ideal.

Amundson and Pontinen employed a matrix inversion procedure, and noted that this procedure is practical only on a very fast computer with a large memory capacity. The method proposed by these authors does not include any forcing or convergence procedures. A bubble-point routine, based on a Newton approximation procedure and normalized stage concentrations, is used to calculate new stage temperatures. The authors state that "the number of iterations depends on the precision required, but generally four or five are adequate". Sargent [24] has found, however, that convergence is often very slow using this procedure, and that sometimes no convergence is achieved at all.

In 1964 Friday and Smith [14] published a paper in which they analysed the reasons for this diverse convergence behaviour. They proposed a set of six decisions, based on physical and numerical criteria, to assist designers in selecting an appropriate procedure for a specific problem. Their work is particularly useful in that it summarizes the work done in the field of computerised solutions to distillation problems, up to that time. The main points to emerge from their study of the situation can be summarised as follows:

(1) No completely general solution procedure had been developed up to that time. They suggested that, owing to the importance of the physical characteristic of different processes, it was unlikely that such a procedure was feasible. A more recent survey has supported this suggestion [18].

(2) The appropriate matching of variables to equations is an important feature of a design procedure.

(3) Procedures involving simultaneous solution of the system equations are superior to stage-to-stage methods in that they can deal with complex columns easily, and are not susceptible to truncation errors.

(4) Stage-to-stage methods are, in general, faster than matrix-type methods and do not require large amounts of computer storage.

The next significant contribution to simultaneous matrix-type solution methods was made by Wang and Henke [25]. They note the work of Friday and Smith, and conclude that the practical advantages of matrix methods outweighed the disadvantages. They criticized previous algorithms on the following points:

(1) They were computationally slow and hence expensive to run.

(2) They required large amounts of core storage which meant they could be run only on big machines.

(3) The use of Newton-Raphson convergence procedures by the algorithms did not guarantee a converged solution. Their work attempts to make matrix-type procedures more attractive.

Wang and Henke adopted the basic method of Amundson and Pontinen, and proposed two alterations in the calculation procedure. These are the use of Gauss-elimination, rather than matrix inversion, to solve the tridiagonal matrix of equations, and the incorporation of Muller's convergence method. Muller's method is described as a generalization of Newton's method of secants, commonly referred to as "reguli falsi", and it is shown to achieve convergence for situations where the Newton-Raphson method diverges.

The Gauss-elimination algorithm has several advantages over the matrix inversion procedure. For large matrices, inversion requires large amounts of both computer storage and time. In addition, it is susceptible to round-off errors, although these are not as significant as in stage-to-stage techniques. The Wang and Henke algorithm is shown to be faster than stage-to-stage methods for complex columns. as well as being inherently more stable for such applications [24]. This algorithm is used in a program published by the CACHE committee. (See Chapter 8.)

Sargent and Murtagh [24] were the first to consider the problem of real stages in connection with a matrix-type solution procedure. They note that Holland [21] had proposed an extension of his stage-to-stage procedure to include efficiency considerations, but dismiss the technique as not being sufficiently rigorous.

Sargent describes a general purpose calculation procedure based on Amundson's original method. The main extension proposed to Amundson's method is a technique for

considering stage performance in terms both of Murphree tray efficiencies and of vapour- and liquid-phase thermal efficiencies. A specially developed relaxation technique is used to achieve convergence, and this is compared to the Newton-Raphson method. The relative stability of the relaxation convergence technique is not discussed, and no mention is made of the Wang-Henke algorithm. A comparison of the computer-time required by the methods of Amundson, Wang, and Sargent indicates that the Wang-Henke procedure is superior in this respect.

A combination of the Amundson matrix method and the stage-to-stage method of Greenstadt et al. [26] was proposed by Naphtali et al. [27]. The algorithm employs a Newton-Raphson convergence technique and can be described briefly as grouping the basic equations by stages, linearizing the equations, and solving simultaneously in tridiagonal matrix form using Gauss elimination. The method is shown to solve problems in which other techniques have failed, although no mention is made of the computer-time required. The grouping of the equations by stages is not recommended by Friday's analysis, as it leads to a build-up of truncation errors, and requires the initial assumption of end compositions.

Greenstadt was the first and only worker to use this grouping method before Naphtali adopted it. The fact that Naphtali's algorithm includes this procedure and the potentially unstable Newton-Raphson convergence technique, to produce an algorithm that will solve both conventional and complex columns, adds support to the theory that specific techniques should be developed for specific problems.

The chief disadvantages quoted by Naphtali are the large core-storage requirements and the need to estimate end compositions. If this estimate is not fairly accurate the method tends to become unstable.

The most recent article surveyed dealing with matrix method is one by Ishii and Otto [28]. They propose a multivariate Newton procedure in which all equations are linearized and solved simultaneously. It is similar to Naphtali's procedure in that it can handle non-ideal solutions. The equations are grouped by type, and the computer storage requirement is reduced by ordering the linearized equations so as to minimise the number of partial derivatives to be evaluated in each iteration.

In conclusion, the following general statements can be made in connection with matrix methods.

(1) They are numerically stable and their accuracy is not significantly affected by round-off errors.
(2) They work equally well for conventional and complex columns.
(3) They require relatively large computer-storage facilities.
(4) The rate and stability of convergence depend largely on the accuracy of initial assumptions and the matching of variables with particular equations.

(b) Class 2 Programs

Programs in this category represent a completely different approach to the solution of distillation problems. A relaxation method is used to consider the dynamic problem of a system settling down to steady-state from an arbitrary initial state, for fixed operating conditions. The method seems attractive because it is equivalent to collecting results from an actual pilot-plant operation, and one might expect the physical stability of such a column to improve the computational convergence characteristics. It has, however, been found that the difficulties encountered in integrating the relevant differential equations are even

greater than those associated with steady-state solution procedures [23].

Rose et al. [29] were the first to publish work in this field. They used a particularly poor numerical procedure, and the rate of convergence was extremely slow. Subsequent investigators used a very stable algorithm and they reported excellent convergence characteristics. The example chosen was later shown to be a particularly favourable one, and it was demonstrated that even with a stable integration technique, convergence was often slow [30]. As a result of these findings the method has been generally ignored by research workers.

Petryschuk and Johnson [31] published the most recent paper on the relaxation method. They compared Rose's technique with Holland's stage-to-stage method, and concluded that the relaxation method was favourable for use in the development of process models and in particular for handling multi-column systems interconnected by recycle streams.

In general, it must be concluded that the dynamic approach inherently requires more computer time, and thus cannot compete with the steady-state methods in solving standard problems.

(c) Class 3 and 4 Programs

Although apparently very different, the Lewis-Matheson [1] and Thiele-Geddes [2] calculation procedures, on which class 3 and 4 programs are respectively based, have much in common. A traditional, though not essential, feature is that the number of trays in the Thiele-Geddes application is kept fixed whereas in the Lewis-Matheson application the number is varied. Thus class 3 programs are usually design-orientated and class 4 are used as rating programs.

(i) The Lewis-Matheson method: Lewis and Matheson [1] were the first to advance a method for solving multi-component systems. Their procedure requires the specification of feed and reflux conditions, the number of plates in each section of the column and an initial estimate of the product distribution. A stage-to-stage calculation procedure is then used to calculate from both ends of the column towards the feed tray. The resulting mis-match at the feed tray is used to re-estimate the product distribution with the procedure being repeated until a match is obtained.

It was not until the appearance of computers that the matching procedure was systematized. Greenstadt et al. [26] were the first to publish a computer technique using the Lewis-Matheson solution method.

They used a Newton-Raphson procedure for the simultaneous solution of the basic equations, which were grouped by stage as mentioned previously. The authors claimed that their method worked both for conventional and for complex columns, but the method was found to be potentially unstable [14]. In addition to Friday's analysis, it was found that unless the initial estimate of the product compositions was reasonable the Newton-Raphson convergence procedure became unstable. This method was not considered again until Naphtali incorporated it in his matrix procedure.

Shortly after Greenstadt's paper, another computerized solution procedure, based on the Lewis-Matheson method, was proposed by Bonner [32]. This algorithm determined the number of trays required for a specified product distribution and reflux ratio. Bonner's work lead to the traditional design orientation of the Lewis-Matheson method. The number of stages in the rectifying and stripping section is adjusted alternately, and thus the feed plate position is finally chosen to minimise the total number of trays required in the column.

The Lewis-Matheson method has, however, not been generally accepted owing to convergence failure when applied to many standard distillation problems.

(ii) The Thiele-Geddes method: Thiele and Geddes proposed a stage-to-stage calculation procedure for multi-component systems that is similar in many ways to the Lewis-Matheson method. An important difference is that the Thiele-Geddes procedure replaces the initial assumption of product distribution, required in the Lewis-Matheson method, with an estimate of the equilibrium ratios for all stages in the column. For systems involving ideal solutions this is equivalent to estimating a temperature profile through the column. The initial estimation of tray conditions required in this method is probably the reason for its traditional role as a rating procedure [24]. It is this factor, as well, that makes the Thiele-Geddes method more stable than that of Lewis and Matheson.

Although the constant equilibrium-ratio profile enhances the stability of the Thiele-Geddes method it makes the procedure susceptible to truncation errors. These errors accumulate in the sections of column between feed and side streams, making the stability of the method unreliable for complex columns [24].

Thiele and Geddes did not indicate how to re-estimate the temperature profile between iterations, and stated that "correction requires much judgment, but one or two repetitions of the calculation will generally give a fairly correct solution to one familiar with the process" [2,24]. However, such a trial-and-error procedure is, in general, very time-consuming. The two most commonly used convergence techniques are Newton's method, and the procedure known as Holland's 'θ' method.

The θ (theta) method was first proposed by Lyster et al. [33], and was successfully applied both to the Thiele-

Geddes and to the Lewis-Matheson procedures. Holland [21] and Petryschuk [31] further developed its application with the Thiele-Geddes solution method.

Holland [21] attempted to extend the Thiele-Geddes-θ procedure to include absorbers and strippers, with limited success. The method was found to be unstable for many absorption problems.

Billingsley [34] was the first to place the θ method on a sound mathematical basis. Seppala and Luus [18] have published a comprehensive comparison of the Thiele-Geddes-θ method and most of the other existing procedures, as mentioned previously.

A recent paper by Holland [35] proposed a modification to the basic θ method that places the technique on a better theoretical basis by using it with a tridiagonal matrix procedure. The new algorithm is shown to be stable for absorber, stripper and reboiled absorber application. This new method is termed the "Multi-θ" procedure. The use of a simultaneous solution procedure rather than the original stage-to-stage method, enables complex columns to be considered.

The application of Newton's method to the Thiele-Geddes calculation procedure was investigated by several workers, including Newman [36], Boynton [37] and Goldstein [38]. Although these methods can solve both distillation and absorber problems more easily than the standard θ method, they have been shown to be considerably slower than the simpler Thiele-Geddes-θ combination [18].

In view of the requirements of the undergraduate distillation course the original combination of the Thiele-Geddes stage-to-stage solution procedure and the Holland-θ convergence technique, proposed by Lyster et al. [33], has been chosen. The reasons for this choice can be summarized as follows:

Summary of the Thiele-Geddes-θ method

(1) The calculation procedure is a stage-to-stage type and can thus be more easily explained to undergraduates. It corresponds directly with the material presented in the course, and requires no knowledge of matrix algebra, or other complex mathematical procedures, in order to follow the computational procedure.

(2) The Thiele-Geddes method, combined with the θ convergence technique, is recognized as one of the fastest computational methods [35]. The Lewis-Matheson procedure, although found to be faster for simple problems, was not considered owing to its unfavourable convergence characteristics.

(3) Non-Ideal mixtures are not considered in the distillation course and are thus ignored here. In addition the data-bank required by a rigorous procedure, such as the Chao-Seader correlation, is not available on the computer system at U.C.T. The Thiele-Geddes-θ procedure is adequate for handling thermodynamically ideal systems.

(4) Matrix procedures, in general, require both more computer storage and more time for execution than do stage-to-stage methods [18,24,25].

(5) As the usual undergraduate tutorial problem seldom involves systems containing more than 5 components and 20 stages, the type of limitation of the Thiele-Geddes method, suggested by Petryschuk [31], is not significant. Modifications to the basic procedure are kept to a minimum, consistent with stability requirements, so that the algorithm can be easily understood at the undergraduate level.

1.25 Recent research trends

In the past, a primary objective of research into process design methods was to reduce capital costs. Since 1973, however, the increased price of fuel has brought the question of operating costs to the fore. In distillation, operating costs can be considerably reduced by running a column at the lower reflux ratios. This requires a very accurate design because product quality is more sensitive to variations in operating conditions at low reflux ratios. Representative system-data are essential to accurate design, with vapour-liquid equilibria and stage-efficiency being two important aspects. Much of the current research on distillation is in these two fields.

In connection with vapour-liquid equilibria, Ellis [5] notes that the limiting factor in many practical designs is the shortage of some binary parameters, necessary for prediction of multicomponent phase equilibria. The most recent treatise in this field, with specific reference to distillation, is Rowlinson's application of the "principle of corresponding states" [39]. Although such advances are of practical value, they are beyond the scope of an undergraduate course. For the purposes of this study Equation (1.1) provides an adequate representation of vapour-liquid equilibria.

Over the years a number of workers have proposed methods for including stage-efficiency in distillation calculations. Most of these use a modification of either Murphree's expression for tray efficiency [40] or Holland's modification of vaporization efficiency. According to Holland [41] vaporization efficiency was first defined by McAdams [42] and Carey [43].

In his first treatment of the subject, Holland [21], observed that vaporization efficiencies were computationally more convenient than the Murphree efficiency factors. Recently Holland and McMahon [41] showed that the Murphree expression

has limitations in that apparently reasonable values of the Murphree tray efficiencies yield negative flowrates within the column. Experimental work by Miskin et al. [23] showed that Murphree tray efficiencies can be negative whereas vaporization efficiencies are always positive. Negative efficiency factors do not necessarily invalidate the Murphree analysis because they are usually associated with large positive values on adjacent trays [5].

Recently a number of models have been proposed that attempt to predict the behaviour of real trays, based on the physical parameters of a system. The main contributors have been Yamada [44,45,46], Holland [44,46], Onda [47] and MacFarland et al. [48]. Asbjørsen [49] has proposed a dynamic model in which vaporization efficiency is re-defined as "equilibrium efficiency". An inefficient stage is considered to be in a state of psuedo-equilibrium with the efficiency factor describing the state of equilibrium. This work has been ignored, probably owing to the inherent difficulties in dynamic solution procedures.

A survey of the available models has been carried out by Standart [50], who is developing yet another model*. He claims that previous models are unrealistic in that they either do not consider the effects of entrainment or attempt to derive overall efficiencies from point efficiency data, rather than from direct mass-balance considerations. A recent paper by Biddulph [51] describes the simulation of an air-distillation column. He uses experimental point efficiency data with an eddy-diffusion model to achieve a good comparison with actual plant experiences. This model assumes, however, that the point efficiency factors at any point in the column are equal for all components [22]. This is true of the argon-nitrogen-oxygen system. For systems not obeying this condition, the direct approach of

*Standart's model still in preparation at time of writing [50].

Standart would be more realistic. At present there is no general method for describing stage efficiency satisfactorily [22].

In this study, vaporization efficiencies are used in the manner prescribed by Holland [21]. It is assumed that experimental data are available to the user in the form described by Miskin [23]. This relatively simple procedure is easily explained at an undergraduate level and provides a fairly realistic description of the behaviour of a real column [22].

1.26 Summary

The literature survey revealed that although a considerable amount of work has been done in the field of multicomponent distillation, the original graphical procedures for binary distillation have not been significantly improved upon.

The programs dealing with binary systems will be developed using the standard graphical solution procedures. The theoretical background will be extracted from textbooks currently prescribed for the course. This will ensure a good correspondence between the course work and the computer package.

The multicomponent program will be based on the Thiele-Geddes calculation procedure, incorporating Holland's θ convergence technique. Only ideal systems will be considered both in the binary and the multicomponent programs. Stage efficiency will be considered in terms of Murphree tray efficiency for binary systems, and vaporization efficiency for multicomponent systems.

Four separate mainline programs will be developed, dealing respectively with:

(a) Binary batch distillation in tray towers - Chapter 2.

(b) Binary continuous distillation in tray towers - Chapter 3.

(c) Binary continuous distillation in packed towers - Chapter 4.

(d) Multicomponent continuous distillation in tray towers - Chapter 5.

Subroutines will be common to all mainline programs requiring the same calculation, thus reducing the overall number of subroutines. The programs will be designed primarily for operation in demand mode with data requests and error messages inserted to guide users.

A program for graphical representation of results will be developed (Chapter 6). This program will incorporate standard "CALCOMP" plotting software and will require a CALCOMP plotter for constructing the various diagrams.

The whole package is designed to supplement the undergraduate course on distillation at U.C.T.

CHAPTER 2

BINARY BATCH DISTILLATION

2.1 INTRODUCTION

Batch distillation is usually employed in industry to treat relatively small quantities of solutions for which the cost of a continuous operation is not justified. It is an unsteady state process, often carried out in a column that is used to distil a variety of solutions. Thus batch distillation problems considered in the undergraduate course entail the rating of a known column for a given feed stock.

A typical batch still with its rectifying column is illustrated schematically in Figure 2.1. A representative McCabe-Thiele diagram is shown in Figure 2.2. The column acts entirely as an enriching section and is therefore described by a single operating line on the x-y diagram. There are two basic modes of operation for batch distillation columns. In one the reflux ratio is increased as the distillation proceeds, thus maintaining a constant distillate product composition. In the other the reflux ratio is held constant and the final distillate composition is determined as the average for the run. The variation in reflux ratio necessary to achieve a constant distillate composition is non-linear and hence sophisticated control equipment is required. Industrial operations are more often carried out under constant reflux.

The theoretical discussion of each mode of operation, given below, assumes McCabe-Thiele conditions of constant molal overflow, and neglects the effects of liquid hold-up within the column. These assumptions are in keeping with

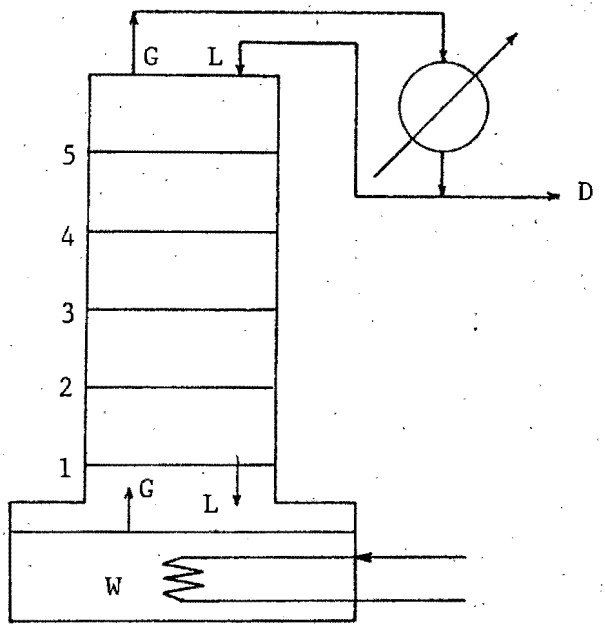


FIGURE 2.1 BATCH DISTILLATION COLUMN

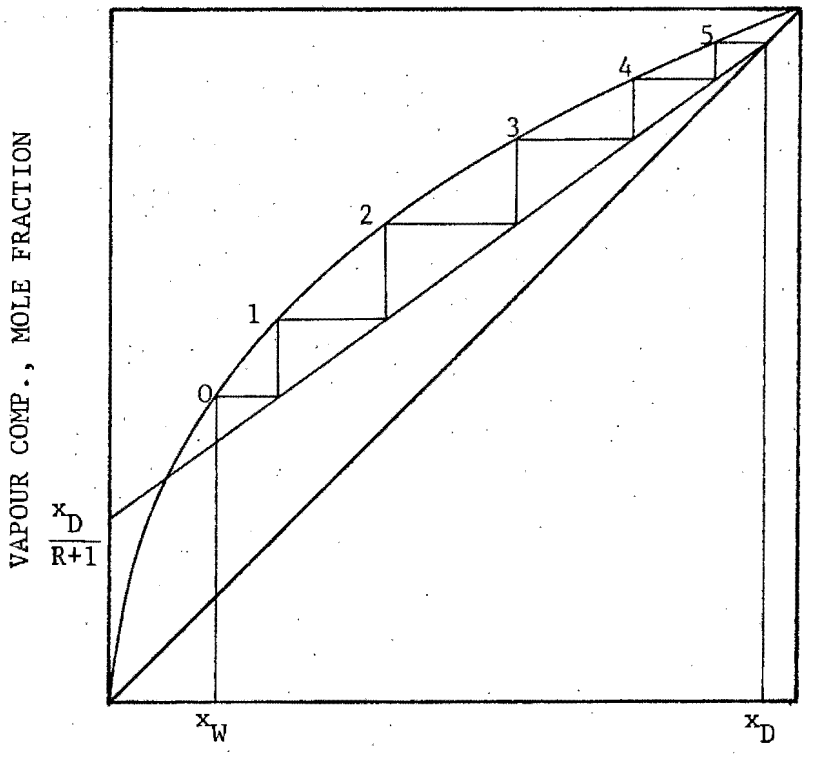


FIGURE 2.2 MCCABE-THIELE DIAGRAM OF BATCH DISTILLATION SYSTEM

the material presented in lectures. In practice this treatment is applicable to systems in which the amount of any component in the batch is at least an order of magnitude larger than the hold-up [52].

2.2 THEORY OF BINARY BATCH DISTILLATION

2.21 Vapour-liquid equilibria

These remarks apply to all the binary programs in the package. The general equilibrium relationship is described by Equation (1.1). For binary systems

$$y_i = K_i x_i \qquad i = 1,2 \tag{2.1}$$

In binary systems the vapour-liquid equilibrium can be described by a curve on an x,y diagram as shown in Figure 2.2. The greater the distance between the curve and the diagonal, the easier is separation by distillation. A useful measure of this distance is the relative volatility, α, defined as

$$\alpha_{1,2} = \frac{y_1}{y_2} \cdot \frac{x_2}{x_1} \tag{2.2}$$

where $y_2 = 1 - y_1$, and $x_2 = 1 - x_1$.

Ordinarily the value of α will change with composition, but the simplifying assumption of constant relative volatility is sometimes made.

Combining Equations (2.1) and (2.2)

$$\alpha_{1,2} = \frac{K_1}{K_2} \tag{2.3}$$

Ideal solution behaviour is assumed throughout the course, and for binary systems Raoult's Law can be expressed, for fixed temperature, as

$$p_i = P_i x_i \qquad i=1,2 \tag{2.4}$$

Also, Dalton's law for ideal vapours is:

$$P_T = p_1 + p_2 \tag{2.5}$$

The equilibrium vapour composition, at the fixed temperature, is thus

$$y_i = \frac{p_i}{P_T} = \frac{P_i x_i}{P_T} \qquad i=1,2 \tag{2.6}$$

Hence

$$K_i = \frac{P_i}{P_T} \qquad i=1,2 \tag{2.7}$$

and

$$\alpha_{1,2} = \frac{P_1}{P_2} \tag{2.8}$$

The vapour pressure of a pure component is expressed as a function of temperature by Antoine's equation

$$\log_{10} P_i = A_i - \frac{B_i}{C_i + T} \qquad i=1,2 \tag{2.9}$$

The user has a choice of three equilibrium data options in the binary programs:

1. The user may specify Antoine constants for each component. The program calculates a set of x,y equilibrium data according to the following scheme: For values of x between 0 and 1,0 in steps of 0,02, the bubble point of liquid mixture x_1, x_2 is computed (see discussion of sub-routine BINBPT). The vapour pressure of the light component at the bubble point is then determined using Equation (2.9). The equilibrium vapour composition is then calculated using

Equation (2.6) and the heavy component value determined by difference ($y_2 = 1 - y_1$). The relative volatility at each point is computed through Equation (2.2). A set of x,y,α data is then stored for use in the solution procedure.

2. A constant relative volatility may be specified. This is acceptable only for McCabe-Thiele systems as it implies a constant temperature through the column. If this option is used, x and y data are computed as required in the program, using Equation (2.2).

3. A set of x,y data, extracted from the literature, or determined experimentally, may be supplied directly. The number of data points must be specified, and to ensure reasonable accuracy at least twenty data pairs should be provided. End points 0,0 and 1,1 are assumed by the program. The relative volatility at each point is computed according to Equation (2.2). If a Ponchon-Savarit system is considered equilibrium temperature data must also be provided under this option. This is discussed in detail in Chapter 3.

2.22 Varying reflux ratio operation

Assuming McCabe-Thiele conditions and negligible hold-up the following equations describe a binary distillation system:

Mass balance

Overall $$F = D + W \tag{2.10}$$

Component $$Fx_F = Dx_D + Wx_W \tag{2.11}$$

Operating line as shown in Figure 2.2

$$y_n = \frac{R}{R+1} x_{n+1} + \frac{x_D}{R+1} \tag{2.12}$$

where R/R+1 is the slope of the line and R is the reflux ratio defined as

$$R = \frac{L}{D} \tag{2.13}$$

Equation (2.12) represents the material balance across tray n in the column. For McCabe-Thiele assumptions L is constant through the column and the operating line is straight.

The following data are usually known when rating a column for a constant distillate composition under varying reflux conditions.

(a) Feed composition x_F
(b) Required distillate product composition x_D
(c) Number of trays in the column N
(d) Final residue concentration x_{Wf}

Assuming perfect trays and referring to Figure 2.2, the first step is to choose a reflux ratio such that the number of steps constructed from x_F to x_D equals the number of equilibrium stages in the system.

Note: The still-pot (a partial reboiler) and a partial condenser, if specified, both represent an equilibrium stage Thus a system with 10 trays and a partial condenser will be described by 12 steps on the McCabe-Thiele diagram.

As product is withdrawn and x_F tends to x_{Wf}, the reflux ratio must be increased to maintain a constant x_D The time required for a batch distillation run can be expressed as follows:

$$\theta_T = F\frac{(x_D - x_F)}{G} \int_{x_{Wf}}^{x_F} \frac{dx_W}{(1-\frac{L}{G})(x_D - x_W)^2} \tag{2.14}$$

This equation was developed by Bogart [53] for systems with negligible hold-up. As the vapour flowrate G is not necessarily known, it is often more convenient to rewrite Equation (2.14) in terms of the total vapour boiled up during a run,V;

i.e. $$V = \theta_T G \tag{2.15}$$

also $$G = L + D \tag{2.16}$$

hence $$V = F(x_D - x_F) \int_{x_{Wf}}^{x_F} \frac{dx_W}{(1-\frac{R}{R+1})(x_D - x_W)^2} \tag{2.17}$$

G is related to the reboiler heat duty and is often limited by the heating device or the column size. V, however, may be determined by graphical or numerical integration and then used to compute either θ_T or G, whichever is unknown.

When solving problems manually, students are required to integrate Equation (2.17) graphically. The program uses a numerical procedure based on Simpson's Rule.

2.23 Constant reflux ratio operation

Varying the reflux ratio in order to maintain a constant distillate product requires a fairly sophisticated control system. It is possible to achieve the same result by holding the reflux ratio constant and collecting a distillate product of varying composition until the average composition reaches a specified value. This requires a higher initial purity than ultimately required, which is not always feasible.

The reflux ratio is fixed throughout the run and the operating lines are parallel to each other on the x-y diagram. A mass balance at anytime can be written as follows:

$$Wx_W = x_D dW + (W-dW)(x_W - dx_W) \tag{2.18}$$

Neglecting second order differentials, Smoker and Rose [54] have shown that for negligible hold-up the quantities of feed and residue may be related by the expression

$$\ln \frac{F}{W} = \int_W^F \frac{dW}{W} = \int_{x_{Wf}}^{x_F} \frac{dx_W}{x_D - x_W} \tag{2.19}$$

The total moles of vapour boiled up, for constant reflux operation is given by

$$V = D(R+1) \tag{2.20}$$

Equation (2.15) can be applied as before to determine either θ_T or G. Equation (2.19) is solved by graphical or numerical integration.

The stepping procedure is performed within the program by alternate solution of the equilibrium and material balance equations. These are expressed in the following form:

Equilibrium (from Equation (2.2)

$$y = \alpha x/[1 + x(\alpha-1)]$$

Material balance (Equation 2.12)

$$y_n = \frac{R}{R+1} x_{n+1} + \frac{x_D}{R+1}$$

The development of the computer program for batch distillation is now discussed. The quoted program text line numbers refer to the printout of the program attached in the Appendices. The variable names extracted from the program are shown in the text in capitals and are defined in the nomenclature (Appendix B).

2.3 DESCRIPTION OF PROGRAM BATCH

2.31 General structure of the program

BATCH is the mainline routine for the solution of batch distillation problems, considering either variable or constant reflux conditions. It shares a common block with only one other subroutine, XDSTEP, although it uses several other routines available to the binary system programs. This

program is different from others in the package in that the mainline is used to read in all data and to output all the results, in addition to performing all the general computation required. Figure 2.3 is an overall flow diagram of the program. Sections requiring further explanation are discussed under the headings of particular subroutines.

2.32 Limitations of the program

BATCH is a rating program in that the number of theoretical stages in the rectifying column is specified by the user.

BATCH is limited to binary systems for which the normal McCabe-Thiele assumptions hold, as discussed in section 2.1. The program is set up to accommodate a maximum of 25 theoretical stages plus a partial reboiler and partial condenser, or 26 stages if a total condenser is specified. A maximum of 55 x,y equilibrium-data points may be specified. The number of data points that may be computed for use in the integration procedure is at present limited to 150. These limits can be adjusted by changing the dimensions of the relevant arrays.

A check on the feasibility of the final reflux ratio must be done by the user. If this is too high the final residue composition should be increased and the problem solved again.

2.33 Operating procedure and data input specifications

BATCH is designed to run on a conversational basis in demand mode. For sample runstreams for solving variable reflux and constant reflux problems refer to Chapter 7. The user may suppress the individual data requests and add a previously prepared data element to save input-output time on the terminal. The data element option should be used for batch mode operation. The conversational responses of

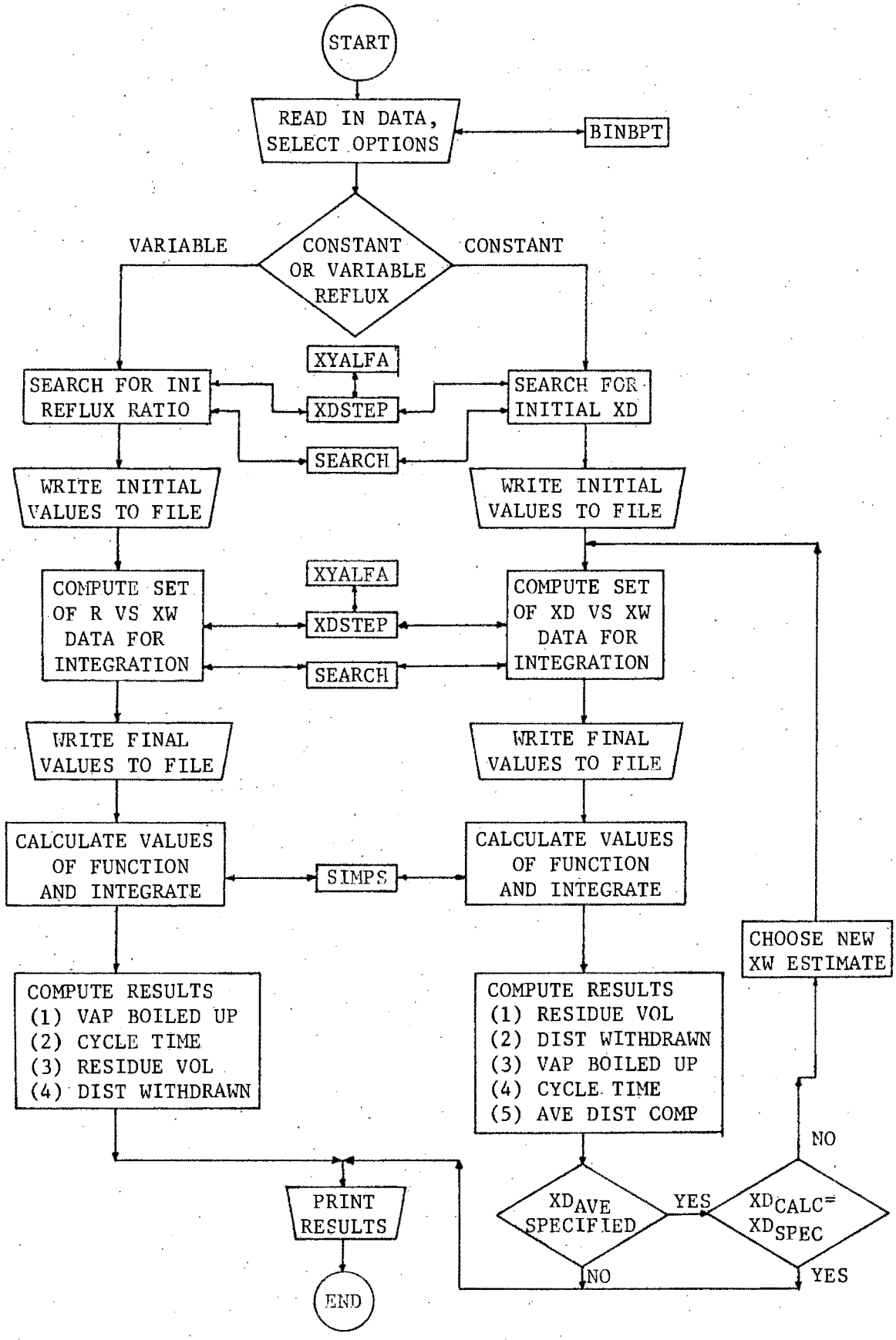


FIGURE 2.3 FLOWCHART OF PROGRAM BATCH

the program in demand mode are shown in the User Manual (Chapter 7). The various options available to the user are also discussed in the Manual.

2.34 Calculation Procedures

2.341 Varying reflux ratio: Once all the necessary data has been accepted, the iterative calculation procedure can be summarised as follows:

Line No.*	Program Action
175,176	Estimate an initial reflux ratio R=1,1 and specify an incremental value to be used in the iterative sequence.
180	Call XDSTEP to determine the residue concentration corresponding to this value of R.
181	Compare resultant residue composition with the specified feed composition. If they are equal, the estimate of R is correct.
183-185	Should the equality not hold R must be adjusted and subroutine SEARCH is used for this purpose. In general, for a fixed distillate composition and number of stages in the column, the greater the reflux ratio the lower the residue composition. Thus counter NCALL causes SEARCH to decrease the value of R if XNTC < XF (see discussion of SEARCH). The procedure is repeated until the convergence limit is met.
(182)	A check on the size of DELR prevents the loop continuing, if for some reason the convergence criterion is not met.
187-192	Resultant initial conditions of reflux ratio and composition are written to a file for use by the plotting routine.

*Note: Line numbers refer to the FORTRAN text shown in Appendix C.

193 The interval (XF - XW) is divided into NINT increments of size H. Present dimension sizes allow a maximum value NINT = 100.

198-218 Computation of data for numerical integration of Equation (2.17). Starting at XW and taking an initial estimate of the final reflux ratio, R=10, calculations between lines 180 and 185 above are repeated for each value of residue concentration from XW to XF in steps of H. The values of XW and R are stored in XDATA and YDATA respectively.

(211-214) Final conditions of reflux ratio are stored for use by the plotting routine.

220-222 Values of the function are calculated at every value of residue composition.

223 Integration of the function from XF to XW is by Simpson's Rule*.

*Note: Numerical integration by Simpson's Rule requires equal intervals along the X-axis. This condition is satisfied by using increment H along the X-axis and fitting a corresponding R to each X value. An alternative procedure would involve finding the initial and final reflux ratios and then dividing the interval $R_{final} - R_{initial}$ into increments. The residue composition corresponding to each value of R would then be determined and a polynomial subsequently fitted to the data. Values of the reflux ratio at equal increments of XW would then be generated from this function. The saving in computer-time involved in reducing the number of search iterations was found to be negligible, and accuracy was lost through the curve-fitting procedure.

224-227 The following data are computed.

(i) Total moles of vapour boiled up during batch cycle - VAP.

(ii) Batch cycle time or vapour boil-up rate, whichever value is not specified initially - TIME or G.

Note: Internally TIME and G are interchangeable. The specific meaning is set by the data input specification and the output record is adjusted accordingly.

(iii) Quantity of residue by material balance - W.

(iv) Total distillate withdrawn - DTOT.

305-345 Results are printed out. For sample printouts see Chapter 8.

2.342 Constant reflux ratio: The basic procedure is similar to that used for varying reflux problems. In these problems, however, either a final residue composition or a final average distillate composition may be specified as an end-point.

Line No.	Program Action
230	Initial estimate of XD taken as 0,99.
234-241	Search for correct initial XD based on specific reflux ratio.
243-248	Initial operating diagram data written to file for use by plotting program.

(a) Specification of final residue composition

The procedure is the same as for varying reflux problems except that a set of XD values are searched for using the constant value of R.

258-282 'DO' loop 21 computes data for numerical integration of Equation (2.19).

(265,266) At constant R, XD decreases with decrease in residue concentration; hence NCALL = 1 and

SEARCH reduces XD if XNTC > XSTEP.

(267,268) Check for XD > 1,0.

284-286 Values of the function to be integrated are calculated. The Simpson Rule routine is used to integrate numerically.

288-292 The following data are computed.

(i) Quantity of residue - W.

(ii) Total distillate withdrawn - DTOT.

(iii) Total moles of vapour boiled up during the cycle - VAP.

(iv) Batch cycle time or vapour boil-up rate - TIME or G.

(v) Average distillate product composition - XDAVE.

Result tables are then printed out as shown in Chapter 8.

(b) Specification of average distillate composition.

If this option is used an initial estimate of the final residue composition is set on line 253 (XW = 0,1). The calculation procedure is then identical up to (v) above, whereupon the calculated average and the specified average are compared (line 294). The SEARCH routine is used to select a better estimate of XW and the procedure is repeated. An error message prevents final residue composition falling below 0,001 (line 297).

2.4 GENERAL COMMENTS ON BATCH

The program is written in standard FORTRAN and is set up for use on the UNIVAC 1106 - EXEC 8 system at U.C.T. Adaptation to other FORTRAN compilers should be accomplished by changing read and write keys, contained in a DATA statement, in the mainline routine.

Units of output variables conform to SI and all flow-rates are on a molar basis. Where the program will accept units other than SI, this is indicated by the input instruction. This is discussed in detail in the User Manual (Chapter 7).

Stages are numbered up from the bottom with the reboiler designated zero. An extra stage is automatically included for partial condenser operation.

Liquid or vapour hold-up and enthalpy considerations are not taken into account.

Error messages include a check on the final residue composition during the adjustment procedure for the average distillate composition specification option, and several data input checks. If XW becomes negative or an input-question reply is incorrect, the program is terminated.

The full operating procedure is outlined in the User Manual (Chapter 7) and test examples are discussed in Chapter 8.

2.5 SUBROUTINES ASSOCIATED WITH MAINLINE BATCH

2.51 Subroutine XDSTEP

This is the only subroutine sharing a common block with 'BATCH'. It is specific to the binary batch distillation program and is used to construct McCabe-Thiele stages on the x-y diagram for batch distillation problems. Although a similar technique is used in the routine STEPD, which performs the stepping operation for the other binary system programs, a separate routine was needed in this case. This is because the batch program is essentially a rating program in which the number of theoretical stages is known, while the other systems are design programs in that they determine the number of stages required. The subroutine is flowcharted in Figure 2.4.

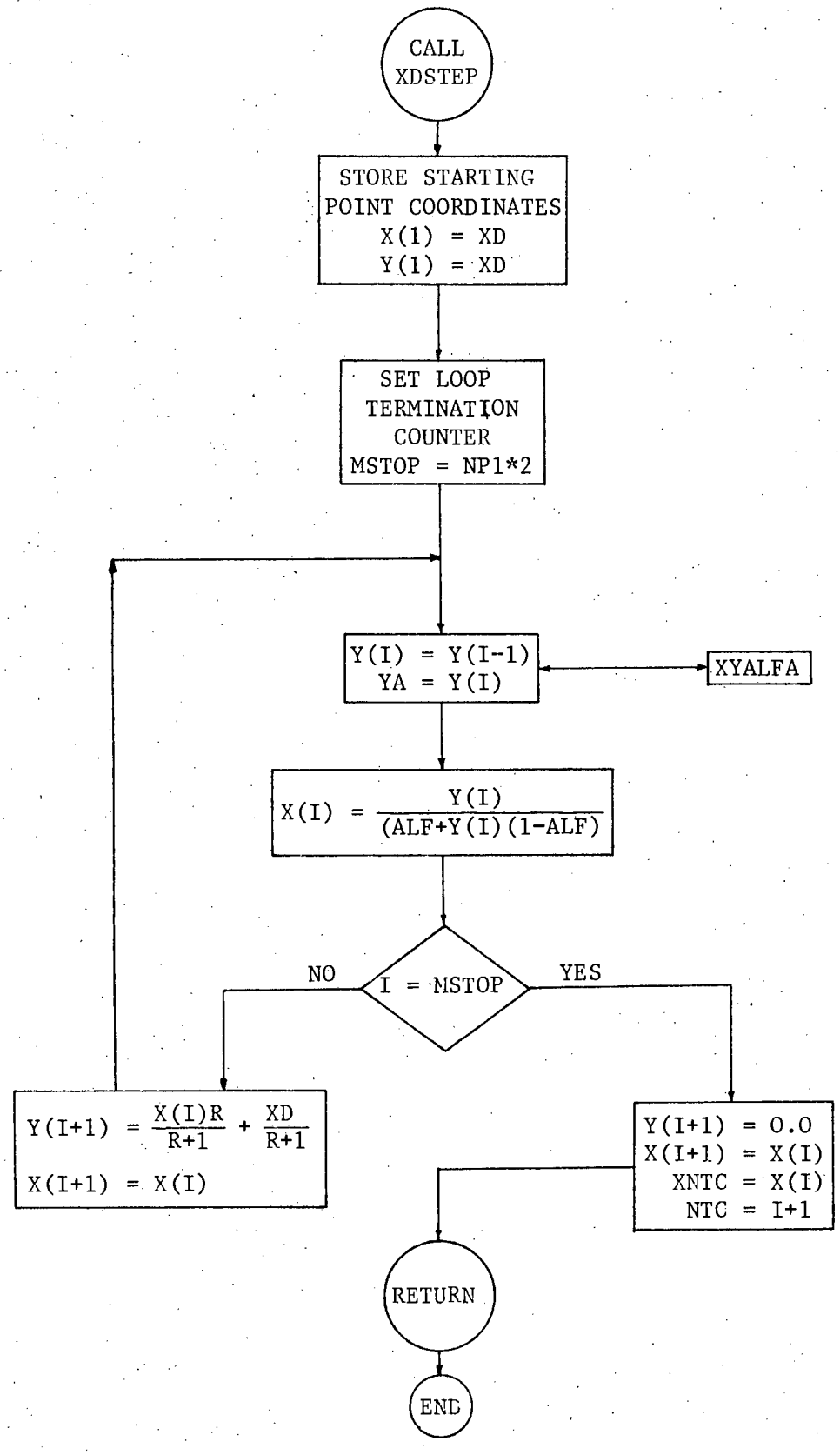


FIGURE 2.4 FLOWCHART OF SUBROUTINE XDSTEP

The computational procedure is equivalent to successive equilibrium - material balance calculations. Starting at point XD on the 45° diagonal an operating line of slope R/R+1 is assumed, and the stages are stepped off as on a McCabe-Thiele diagram. The coordinates of each point determined in the equilibrium curve and operating line are stored in arrays X and Y. Counter NTC records the total number of such points for use in the plotting routine. The sequence of calculations is shown diagramatically in Figure 2.5 below.

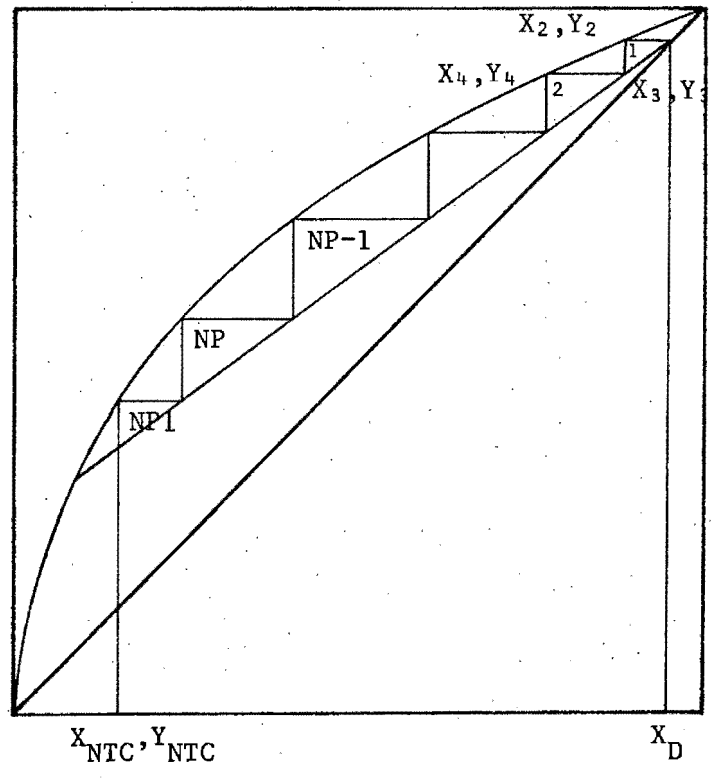


FIGURE 2.5 STEPPING OFF STAGES BY SUBROUTINE XDSTEP

Stage numbers shown in Figure 2.5, 1,2,..., NP1 are those used in the routine. The resulting printout and x-y diagram plotting data are converted to the conventional numbering system which designates the reboiler as zero and numbers from the bottom upwards.

Calculation procedure:

Line No.	Program Action
7,8	Store starting point XD as X(1),Y(1).
11-13	Store Y(2) = Y(1) = XD and call subroutine XYALFA to determine corresponding relative volatility ALF.
14	Compute X(2) using variation of Equation (2.2).
16	Compute Y(3) from material balance expression (operating line equation).
17	Store X(3) = X(2).

Counter MSTOP terminates loop after NP1 stages have been stepped off. The final material balance calculation, representing the reboiler is stored as X(NTC), Y(NTC) with Y(NTC) = 0 for convenient x-y diagram construction by the plotting program.

XNTC ≡ X(NTC) is returned to the mainline for comparison with the set value of the residue composition.

2.52 Bubble point and Dew point calculations

2.521 Subroutine BINBPT: This subroutine has no common block and is employed by all the binary programs. Its main function is to return to the calling program the bubble point of the liquid solution with composition X1. It is also used in the computation of equilibrium data when Antoine constants are specified (see section 2.21). For this purpose the vapour pressure, P1, of the light component is returned.

Two procedures for computing the bubble point of a solution are included. The choice is determined by the type of equilibrium data specified.

(a) Antoine constant specifications

For such systems the routine employs a standard Newton procedure to converge on a bubble point T1 at composition X1. The Antoine constants for each component, the total pressure and an initial estimate of T1, are provided by the calling program through the calling list. The specified pure component boiling point is used as an initial estimate of T1. Newton's method is used to converge Raoult's law to find T such that $\Sigma y_i = 1$.

From Raoult's Law $y_i P_T = P_i x_i \qquad i = 1,2$

$$P_T \Sigma y_i = \Sigma P_i x_i$$

At bubble-point $\Sigma y_i = 1,0 \qquad i = 1,2$

hence $$P_T = \Sigma P_i x_i$$

or $$P_T - \Sigma P_i x_i = 0$$

where P_i is given by Antoine's function of temperature.

In Newton's method, a better estimate of the temperature is related to the previous estimate by the equation

$$T_{n+1} = T_n - \frac{f(T_n)}{f'(T_n)}$$

where $f(T) = SUM1 = P_T - \Sigma P_i x_i$ and for $P_i = 10^{(A_i - \frac{B_i}{C_i+T})}$

the first derivative $f'(T) = SUM2 = -2,303 \frac{B_i}{(C_i+T)^2} \cdot P_i x_i$

Convergence is reached as $\frac{SUM1}{SUM2} \rightarrow 0$

Once the bubble point has been determined in 'DO' loop 10, the vapour pressure of the light component, P1, is calculated at the bubble point (line 24).

(b) x-y-T data specification

This procedure is used in Ponchon-Savarit designs when equilibrium composition and temperature data have been supplied. The bubble point is determined by searching through the available data. A straight line interpolation is used between data points and for NDATA $\geq$ 20 this procedure is satisfactory for most systems. For systems that deviate from Raoult's law, such as acetone-water, more points should be provided.

2.522 Subroutine BINDPT: A variation of BINBPT is contained in BINDPT for computing dew points for binary systems. The only difference between the two is the structure of the equations used in the Newton convergence procedure. The two methods were not included in one subroutine because both calculations are not always required. Developed as separate routines, they are available for use in other mianline programs requiring either bubble or dew point calculations.

2.53 Subroutine XYALFA

This subroutine has no common block and is used by all the binary system programs. The routine is used to search for a value of the relative volatility ALPHA corresponding to the vapour or liquid composition value XY1 supplied by the calling program. For systems with constant relative volatility the routine returns the constant value ALFCON. Linear interpolation is used between actual data points, stored in ALFA, which means that accuracy is better for a greater number of points. It has been found that for most reasonably ideal systems a minimum of 20 data points is acceptable. The calling program determines whether

XYALF and XY1 represent vapour or liquid compositions and the corresponding relative volatility value is returned as ALPHA. NDATA is the total number of data points available and is the size to which ALFA and XYALF are automatically expanded. Counter MALF is set equal to 2 if a constant relative volatility is specified. A DATA statement in the mainline sets MALF = 1 initially.

2.54 Subroutine SIMPS

This subroutine is used by both BATCH and PACKED for numerical integration of various expressions. The technique used is that known as Simpson's Rule.

Consider a function f(x) divided into n increments of size h between limits a and b. For an even number of increments, f(x) is expanded by Simpson's analysis, to give:

$$SUM = \int_b^a f(x)dx = \frac{h}{3}[(f_1+4f_2+f_3)+(f_3+4f_4+f_5)+\ldots + (f_{n-1}+4f_n+f_{n+1})]$$

$$= \frac{h}{3}[f_1+4f_2+2f_3+4f_4+2f_5+\ldots+2f_{n-1}+4f_n+f_{n+1}]$$

Hence

$$SUM = \frac{h}{3}\sum_{i=1,3,5,\ldots}^{n-1}[f_i+4f_{i+1}+f_{i+2}]$$

Subroutine SIMPS applies this formula to function values f_i, stored in FUNC, over M intervals of size H between limits ALIM and BLIM. The value of the resulting integral is returned to the calling routine as SUM.

2.55 Subroutine SEARCH

This routine is a standard iterative calculation technique, and is used in most of the iterative procedures in the package.

In general FA1 ≠ FA2 is to be converged to FA1 = FA2 where FACTOR is a variable that directly or indirectly affects the equality. DEL is the change in FACTOR per iteration that is reduced as convergence is reached. Counter MM causes DEL to be reduced as FA1 > FA2 switches to FA1 < FA2 and vice versa. N defines the overall effect of FACTOR on the inequality.

Example of use of SEARCH.

Consider a batch distillation system operating under variable reflux.

Distillate composition XD is fixed and the initial reflux ratio required for a feed of composition XF is unknown.

An estimate of reflux ratio R is made and a stepping procedure is used to step off the given number of stages from XD. The resultant residue composition will not in general equal XF. Let FA2 equal XF and FA1 equal the resultant residue composition. FACTOR is the reflux ratio and DEL, an incremental value. If the initial trial results in FA1 < FA2, the reflux ratio must be reduced to improve the result, hence N = 2 for this procedure.

The variables used in this subroutine are not defined in the nomenclature. All the variables are contained in the calling list and initial values are provided by the calling program.

2.6 SUMMARY OF BINARY BATCH DISTILLATION PROGRAM

Program name: BATCH
Type: Mainline with subroutines
Operation mode: Demand or Batch on UNIVAC 1106 - Exec 8 system.

1. Title: Batch Distillation of Binary Systems by McCabe-Thiele method.

2. Solution type: Rating program.

3. Application: BATCH can be used to solve problems involving batch distillation systems operating under either variable or constant reflux conditions. The program is limited to systems for which standard McCabe-Thiele assumptions hold. It is essentially a system rating program in that it requires the number of theoretical stages in the rectifying column to be specified. The program is limited to 25 trays, although this number can be increased if required.

The user must provide equilibrium data in the form of x,y data, Antoine constants for each component, from which x,y data are then computed, or a constant relative volatility value.

Important features of BATCH include a Newton method bubble point routine for computing equilibrium data from Antoine constants, an iterative search technique used to generate data for the necessary numerical integrations, and a Simpson's Rule numerical integration routine.

4. Program Development:
   (a) Computer used UNIVAC 1106 - Exec 8
   (b) Minimum Core requirements: 10 K
   (c) Language - Standard FORTRAN IV
   (d) Typical run time: Variable reflux: 7 secs
                         Constant reflux: 9 secs

CHAPTER 3

BINARY CONTINUOUS DISTILLATION - TRAY COLUMNS

3.1 INTRODUCTION

True binary distillation problems do not occur very often in practice, but despite this they are very useful for demonstrating the basic principles of the process. Their usefulness derives from the fact that such problems are conveniently solved graphically, in two dimensions. The diagrammatic representation provides the student with a means of visualizing the design and performance of a distillation column.

All three aspects of distillation column design mentioned in Chapter 1 are considered for binary continuous distillation systems. They are:

(1) Vapour-liquid phase equilibrium on each stage.
(2) Material and energy balances around each stage.
(3) Stage efficiency.

The physical dimensions of the column are not considered.

Unlike batch distillation, continuous distillation may be regarded as a steady-state operation. The two graphical design procedures commonly used were proposed by

(a) McCabe and Thiele, and
(b) Ponchon and Savarit.

The McCabe-Thiele procedure assumes:

(a) that the molar latent heats of the two components are equal;

(b) the heat capacities are constant over the temperature range, and

(c) that there is no heat of solution.

Enthalpy effects are thus ignored, constant molal overflow is implied and the operating lines on the x-y diagram are straight. This design procedure, although not rigorous, is adequate for systems of chemically similar components.

The Ponchon-Savarit procedure, by comparison, is rigorous and can be applied to all binary systems considered in the undergraduate course. Detailed enthalpy data must, however, be supplied.

The computer program developed here includes both techniques. The user specifies the procedure required and enteres data accordingly. Each procedure will be discussed separately.

3.2 THEORY OF BINARY CONTINUOUS DISTILLATION

3.21 McCabe-Thiele design

The basic simplifying assumption that characterises the McCabe-Thiele procedure is that of equimolal overflow, as defined above. A schematic diagram of a continuous distillation column and the corresponding McCabe-Thiele diagram are shown in Figure 3.1(a) and (b).

The column shows one feed stream, no side streams, a partial reboiler and a total condenser. For systems in which the heavy component is water, the partial reboiler may be replaced by open steam. The total condenser may be replaced by a partial condenser with either a single phase or a two phase distillate product withdrawn. The McCabe-Thiele diagrams for these options are shown in Figures 3.2(a) and (b).

Finally, the column may include several feed and side streams. A McCabe-Thiele diagram for a column with two feeds and a side stream positioned between the feeds is shown in Figure 3.2(c).

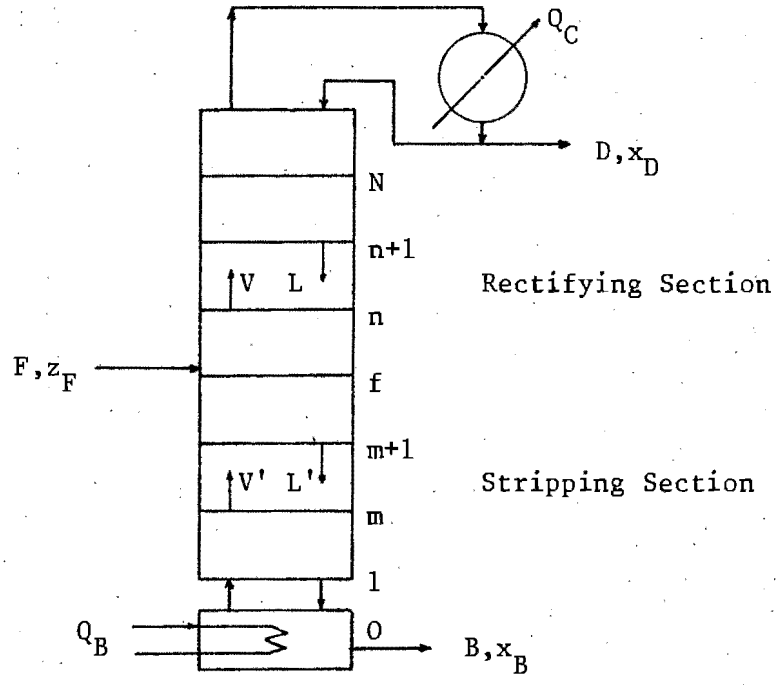


FIGURE 3.1(a) CONTINUOUS DISTILLATION COLUMN

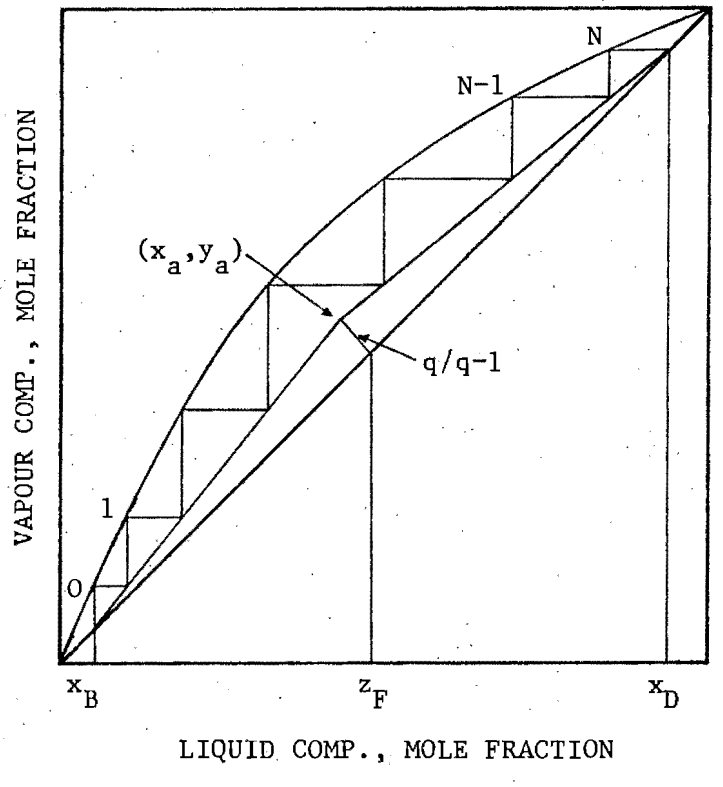


FIGURE 3.2(b) MCCABE-THIELE DIAGRAM OF CONTINUOUS DISILLATION SYSTEM

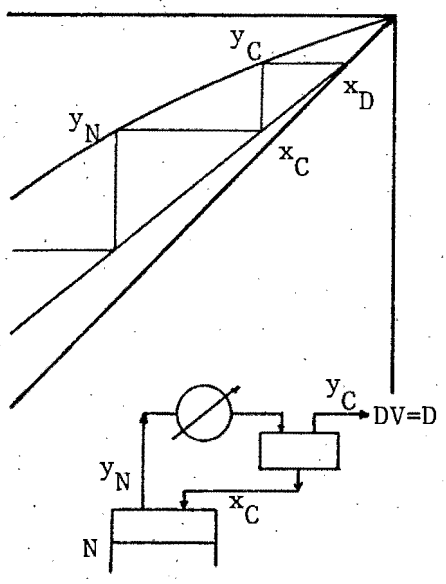


FIGURE 3.2(a) PARTIAL CONDENSER
ALL VAPOUR DISTILLATE

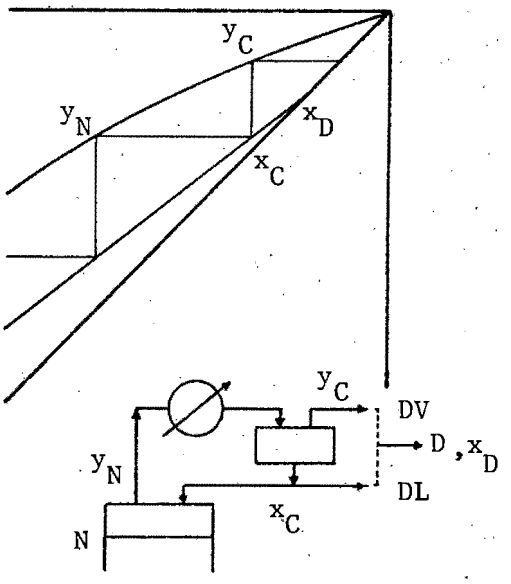


FIGURE 3.2(b) PARTIAL CONDENSER
VAPOUR & LIQUID DISTILLATE

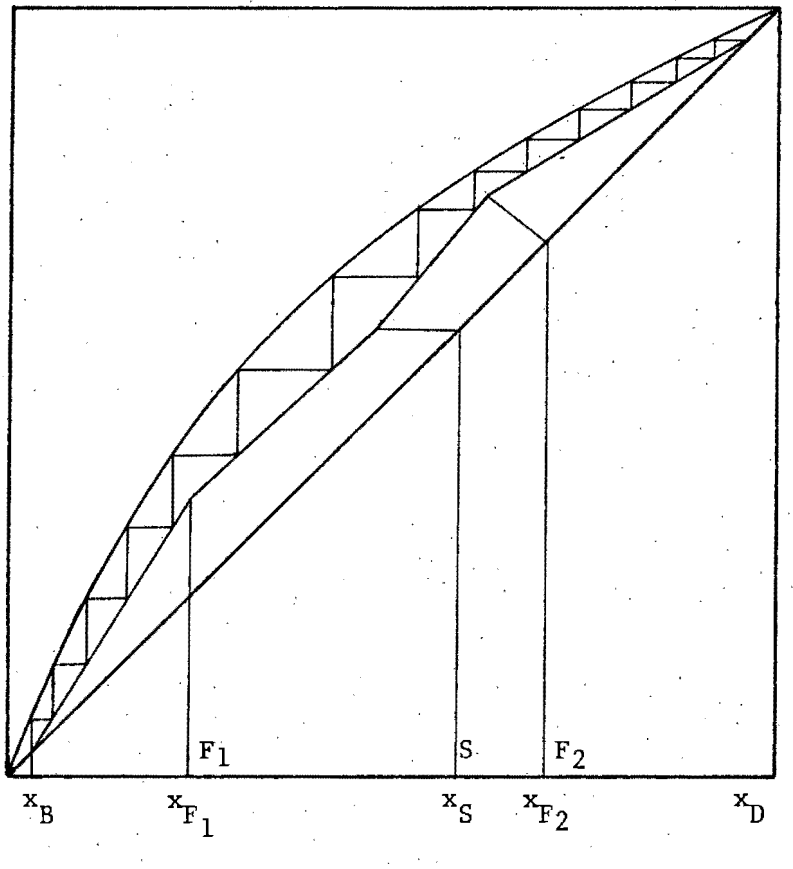


FIGURE 3.2(c) COMPLEX COLUMN
2 FEEDS AND 1 SIDE-STREAM

The McCabe-Thiele graphical design of a simple column as shown in Figure 3.1 is described by the following equations.

1. Vapour-liquid equilibrium.

The equations listed in section 2.21 apply to continuous systems.

2. Material Balance.

On applying the McCabe-Thiele assumptions the general material balance equation (1.2) can be written around each section of the column.

Rectifying Section:

Overall balance $V = L + D$ (3.1)

Component balance $Vy_n = Lx_{n+1} + Dx_D$ (3.2)

Resulting operating line,

$$y_n = \frac{R}{R+1} x_{n+1} + \frac{x_D}{R+1} \quad (3.3)$$

Stripping section:

Overall balance $L' = V' + B$ (3.4)

Component balance $L'x_{m+1} = V'y_m + Bx_B$ (3.5)

Overall column material balance

$$F = D + B \quad (3.6a)$$

Component balance

$$Fz_F = Dx_D + Bx_B \quad (3.6b)$$

For complex columns the overall feed is computed as the difference between the sum of all feeds and the sum of all side streams. The Bottoms flowrate can then be calculated using Equation (3.6).

3. Stage efficiency in terms of Murphree tray efficiency factors.

$$E_M = \frac{y_{n+1} - y_n}{y^*_{n+1} - y_n} \quad (3.7)$$

The thermal condition of the feed is not generally known and should be determined by a flash calculation. For McCabe-Thiele systems, enthalpy and equilibrium-temperature data are not specified and hence a flash calculation is not possible. The feed condition is thus defined as the fraction of liquid in the feed, q, such that qF are the moles of liquid in the feed. The q-value must be known in order to determine the point of intersection of the operating lines. The locus of these points of intersection can be expressed as

$$y_q = \frac{q}{q-1} x_q - \frac{z_F}{q-1} \tag{3.8}$$

Equations (3.3) and (3.8) can be solved simultaneously to define the coordinates of the point of intersection of the operating lines (see Figure 3.1(b)).

$$x_a = [z_F(R+1) + x_D(q-1)]/(R+q)$$
$$y_a = [z_F R + x_D q]/(R+q) \tag{3.9}$$

The manual design procedure involves constructing the x-y equilibrium curve, inserting the operating lines and stepping off the number of stages as shown in Figure 3.1. The stepping procedure will not in general, link x_B and x_D exactly. The fraction of a stage represented by a miss-match is computed as

$$\text{Stage fraction} = (x_D - y_{n-1})/(y_n - y_{n-1}) \tag{3.10}$$

The first step on the x,y diagram represents the reboiler and this is subtracted from the total to determine the number of theoretical trays in the column. Another step must be subtracted if a partial condenser is specified, see Figure 3.2(a & b).

3.22 Ponchon-Savarit design

The detailed development of the graphical design procedure first proposed by Ponchon and Savarit [8] is set out in Treybal [9], Smith [11] and several other texts on binary distillation.

The basic equations defining a binary system have been shown in section 3.21. When McCabe-Thiele assumptions are not made, an energy balance must be considered. In addition, flowrate subscripts must be included in the material balance equations to identify the individual stage positions.

The general enthalpy balance Equation (1.3) can be written around each section of the column as follows. Heat losses from the column are assumed to be negligible.

Rectifying section:

$$V_n H_n = L_{n+1} h_{n+1} + DH_D + Q_C \tag{3.11}$$

Stripping section:

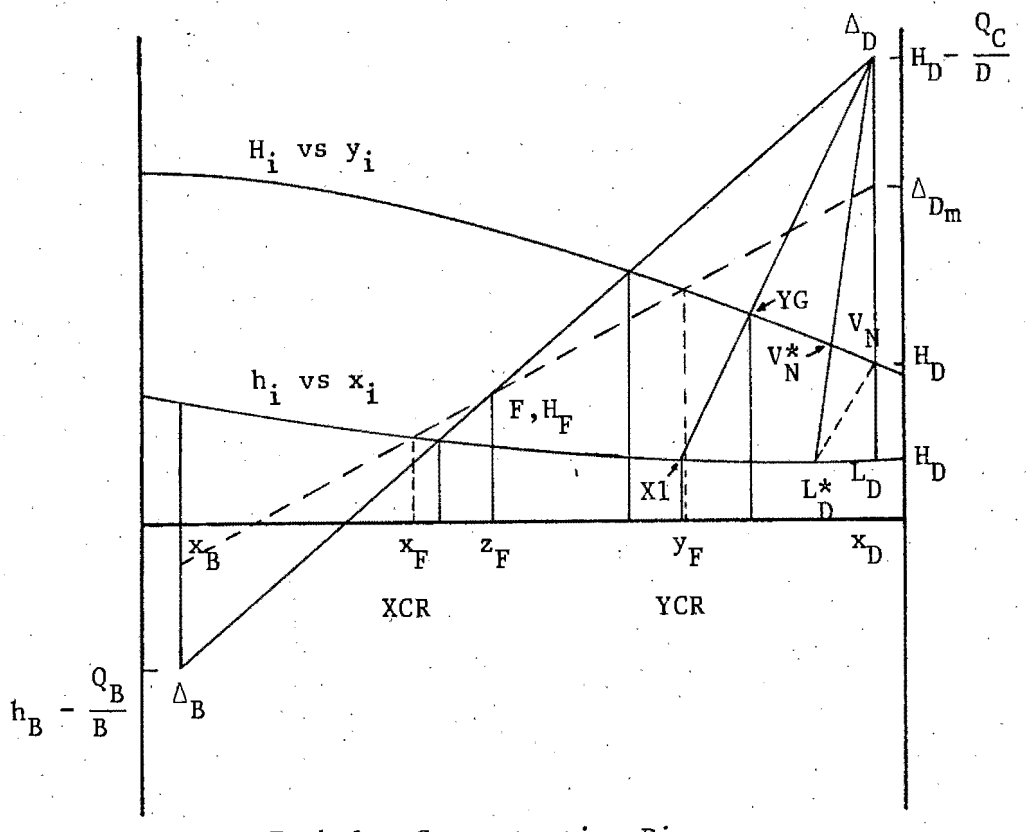
$$L_{m+1} h_{m+1} = V_m H_m + Bh_B - Q_B \tag{3.12}$$

Equations (3.2) and (3.4) are rewritten as follows to account for varying flows through the column

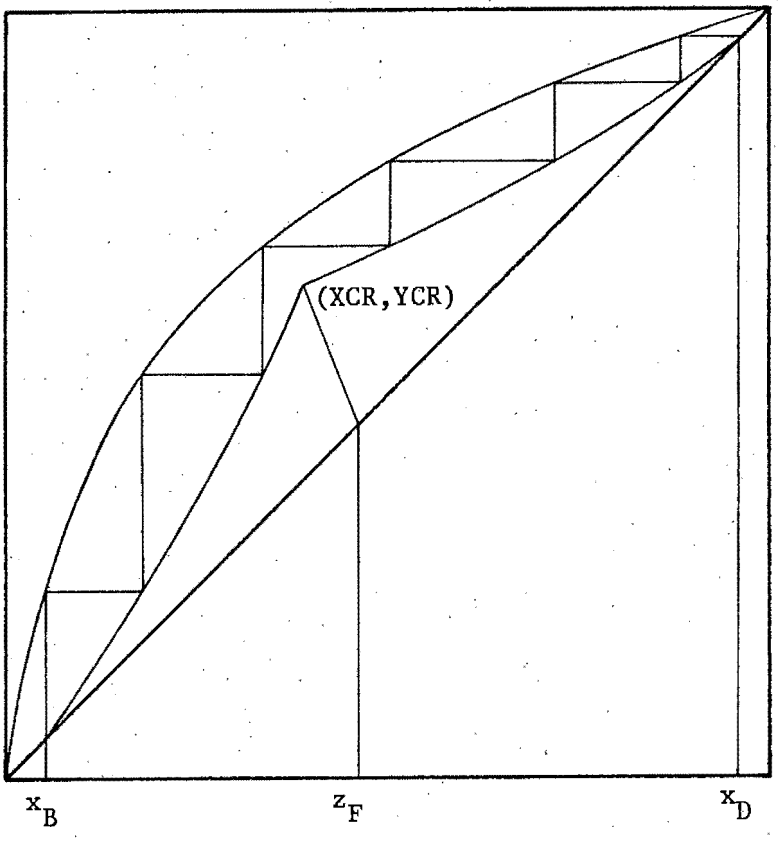
$$V_n y_n = L_{n+1} x_{n+1} + Dx_D \tag{3.13}$$

$$L_{m+1} x_{m+1} = V_m y_m + Bx_B \tag{3.14}$$

The operating lines are not straight and the material and energy balances must be solved simultaneously with the equilibrium expression. The Ponchon-Savarit procedure solves the equations graphically. Figure 3.3 shows an enthalpy-concentration diagram and the corresponding x,y diagram for a binary system. The first step in a Ponchon-



Enthalpy-Concentration Diagram



x,y diagram

FIGURE 3.3 PONCHON-SAVARIT DIAGRAM

Savarit design is the development of the enthalpy-concentration diagram which is then used to construct the operating lines on the x,y diagram.

The Ponchon-Savarit operating diagram is used in the design both of tray columns and of packed columns in this study. The program for designing packed columns (see Chapter 4) is thus referred to in the description of the computer techniques used to construct this operating diagram.

(a) Enthalpy concentration diagram.

The construction of the enthalpy-concentration diagram as described below is shown in Figure 3.3.

The enthalpy-concentration curves are established, assuming ideal solution behaviour using the following relationships.

Enthalpies of streams leaving stage j

$$\text{liquid enthalpy} \quad h_j = x_1 h_1 + x_2 h_2 + H_S \tag{3.15}$$

$$\text{vapour enthalpy} \quad H_j = y_1 H_1 + y_2 H_2 \tag{3.16}$$

Pure component enthalpies h_i and H_i are computed at the solution bubble point and dew point temperatures respectively. H_S, the heat of solution, is included only if such data are available. It is usually convenient to adopt the temperature, at which heat of solution data are quoted, as the base temperature for enthalpy calculations.

Pure component enthalpies are computed as follows:

$$h_i = \int_{T_b}^{T_B} CpdT \tag{3.17}$$

$$H_i = \int_{T_b}^{T_D} CpdT + LH_{i,T_D} \tag{3.18}$$

where LH_{i,T_D} is the latent heat of vaporization of the component, calculated at the dew point temperature of the solution.

The heat capacity, Cp, is expressed as some function of temperature for which coefficients are supplied. The following function, developed by Kobe [55] is used in the programs in this study.

$$Cp_i = a_i + b_iT + c_iT^2 + d_iT^3 \qquad T \text{ in } {}^{o}K \tag{3.19}$$

The latent heat of vaporization for each component at its boiling point under column pressure must be supplied by the user. These values are extrapolated to other temperatures using Watson's correlation [56], which is

$$LH_{i,T} = LH_{i,T_{BP}} \cdot \left[\frac{T_C - T}{T_C - T_{BP}}\right]^{0,38} \tag{3.20}$$

This correlation requires component critical temperature data.

The overall heat balance around the column can be expressed as follows

$$FH_F = DQ' + BQ'' \tag{3.21a}$$

or

$$F = \Delta_D + \Delta_B \tag{3.21b}$$

where Δ_D and Δ_B represent imaginary streams defined as

$$\Delta_D = DH_D + Q_C \tag{3.22}$$

$$\Delta_B = BH_B - Q_B$$

Thus the "enthalpy" of these streams includes the condenser and reboiler heat duties respectively. The overall heat

and material balance is represented on the diagram by line Δ_D - F - Δ_B (from Equations (3.21b)).

Lines $\Delta_D L_D$ and $\Delta_D L^*_D$ represent the material and enthalpy balance around a total and partial condenser respectively. By the lever-arm rule, the reflux ratio for a total condenser is given as:

$$R = \frac{\text{line } \Delta_D V_N}{\text{line } V_N L_D} \tag{3.24}$$

and similarly for a partial condenser.

The tie line passing through point F on the diagram is defined by the equilibrium expression for the system. If this line is extended to points Δ_{D_M} and Δ_{B_M} the separation thus defined requires an infinite number of stages in both sections of the column. This pinch point condition at the feed position enables the minimum reflux ratio to be calculated as:

$$R_M = \frac{\text{line } \Delta_{D_M} V_N}{\text{line } V_N L_D} \quad \text{(for total condensers)} \tag{3.25}$$

The program uses this technique for computing R_M, whereupon it checks that the specified value is greater than the minimum. If $R_{spec} < R_M$ a value of $R = 1{,}5\ R_M$ is assumed and the calculations are continued. Treybal [9] notes that for many systems the optimal reflux ratio lies between 1,2 R_M and 1,5 R_M. This procedure avoids pinch points at the intersection of the operating lines. Pinch points resulting from Raoult's law deviations are determined during the stepping procedure.

To calculate R_M as described above, the tie line through F must be known. For Ponchon-Savarit designs it is assumed that the feed is at the same pressure as the column and that the temperature is known. The feed condition can be expressed

in terms of q which is defined as the heat required to convert 1 mole of feed from its condition H_F to a saturated vapour [9]. The q-value is thus defined by the equation

$$q = \frac{H_{V_F} - H_F}{H_{V_F} - H_{L_F}} \tag{3.26}$$

where H_{V_F} and H_{L_F} are the enthalpies of the feed as a saturated vapour and liquid respectively.

For superheated or supercooled feed the above equation is used in the program to calculate q. If a two phase feed is considered, Equation (3.8) is written as

$$q = \frac{y_q - z_F}{y_q - x_q}\ , \tag{3.27}$$

and is used as described in section 3.53. At this stage sufficient information is available to construct the operating lines on the x,y diagram.

(b) x,y diagram.

The equilibrium curve on the x,y diagram is determined as described in section 2.21. The manual procedure described in textbooks constructs the operating lines and the stepped stages simultaneously. It is more convenient to perform these constructions separately on a computer because, if the operating lines are defined, the same basic stepping procedure may be used both for McCabe-Thiele and for Ponchon-Savarit designs. In addition, the design of a packed column requires only the operating lines. Thus the same computer technique can be used for both types of column if stepping is not done simultaneously.

The program computes the coordinates of a number of points defining the operating lines for each column section. These data are stored in arrays for use in design of either tray or packed columns. The same stage-stepping subroutine is used in both design procedures for determining the number

of trays in the column. A different calculation technique is used in each case, however, because the operating line is defined by a series of points in Ponchon-Savarit design, and by a straight line equation in McCabe-Thiele designs. (See sections 3.516 and 3.517.)

3.3 DESCRIPTION OF PROGRAM BINARY

3.31 General structure

BINARY is the mainline program for solving binary continuous distillation problems using either McCabe-Thiele or the Ponchon-Savarit design procedure. The mainline program is used primarily to link the various subroutines together for a specified design procedure. All the calculations are performed in these subroutines with the mainline providing the relevant data and determining the calculation sequence. The overall operation of BINARY is flowcharted in Figure 3.4.

3.32 Limitations

The program is a design facility in that it computes the number of trays required to achieve a specified product purity. A Murphree tray efficiency factor may be included to account for real tray behaviour. McCabe-Thiele systems involving simple or complex columns may be considered whereas Ponchon-Savarit designs are restricted to single columns only. It is assumed that all binary systems considered are ideal and that, for Ponchon-Savarit designs, enthalpy function coefficients are available in the correct form. Entrainment, weeping and heat losses are not considered in either case.

Either partial or total condenser may be specified for either design procedure. Open stream operation may replace the partial reboiler in McCabe-Thiele problems. A flash calculation determines feed condition for Ponchon-Savarit designs. The feed q-value must be specified directly for

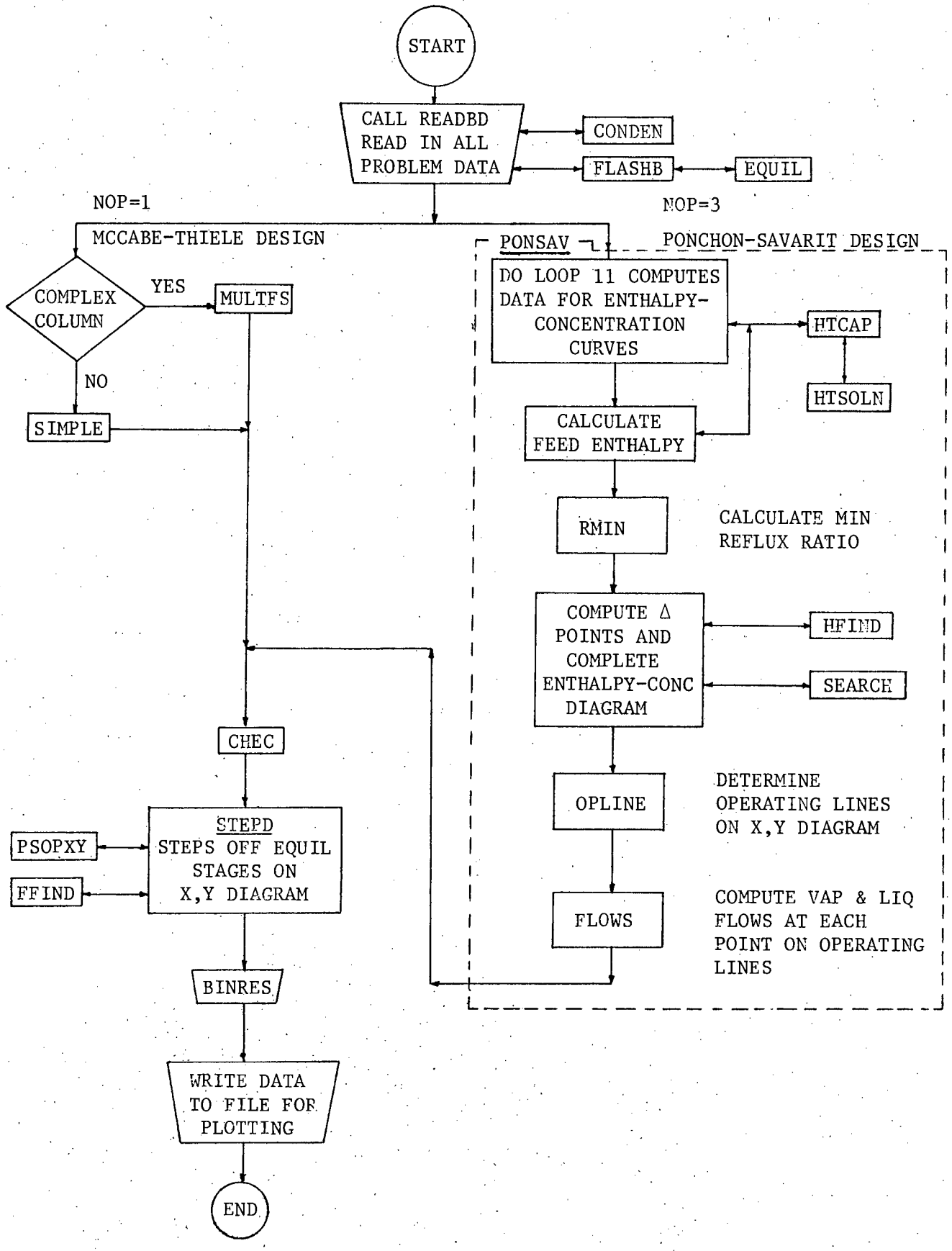


FIGURE 3.4 FLOWCHART OF PROGRAM BINARY

McCabe-Thiele designs. The dimensions of the column are not computed in either case (see Chapter 8). A full description of the options available to the user is given in the User Manual, Chapter 7.

3.33 Operating procedure

The program is designed for operation from either a terminal in demand mode or via a card deck in batch mode. A conversational facility is available in demand mode but, because of the type of enthalpy data required, the user is expected to insert this in a data element when using the Ponchon-Savarit design option (User Manual). The conversational output may be suppressed and a complete data element used.

Data for plotting operating diagrams for either design procedure are written to file by BINARY for subsequent use by the CALCOMP plotting program. Data files must be assigned for this purpose before BINARY is executed, if plotting is required. Subroutine TEACH provides a brief description of the program's capabilities and then terminates the run because users who are not familiar with the program are unlikely to operate it successfully without preparation.

3.34 Calculation procedure in mainline program BINARY

Line No.*	Program Action
5,6	DATA statement for defining constants and counters.
7,8	NPPSC and NPTCC are limits used to prevent dimension overflows in various calculation sequences. NPPS and NPTC are parameters defining the size of arrays (see COMMON block BINCOM).
9-32	Program determines the design method required by the user and the data input procedure to be used.

*See FORTRAN text, Appendix C.

33 Program calls subroutine READBD which requests data conversationally (MCAN=1) or reads a data element (MCAN=2). These options are described in the User Manual, Chapter 7.

(a) McCabe-Thiele design procedure.

36-39 If a complex column is specified, subroutine MULTFS is called to set up operating lines and compute flowrates in each section of the column (see section 3.55). Control is then transferred to line 49 for stepping off stages on the x-y diagram.

40-44 Subroutine SIMPLE determines the operating lines and section flowrates for simple columns. (See section 3.56.) Control is then transferred to line 49 as above.

(b) Ponchon-Savarit design procedure

46-48 Subroutine PONSAV constructs the complete operating diagram consisting of an enthalpy concentration diagram and an x,y diagram. The heat duties of the condenser and reboiler are also computed (see section 3.57). On return to the mainline the run continues as for McCabe-Thiele designs.

(c) Stepping off equilibrium stages on the x,y diagram

49 302 CONTINUE

50,51 Subroutine CHEC ensures that no pinch points exist on the x,y diagram (see section 3.515).

52 Subroutine STEPD steps off the stages and computes the total number of theoretical trays required in the column (see section 3.516).

53 Subroutine BINRES prints out the table of results. The user may truncate the table if the full set of results is not required. (See section 3.519.)

54-85 Data for use by the plotting program is written directly into previously assigned files. This is described in the User Manual.

87-89 An error message is printed out if errors are detected in the user's conversational inputs. Other error messages are contained in the associated subroutines.

3.4 GENERAL COMMENTS ON BINARY

The linear interpolation search procedure described in section 2.5 (subroutine XYALFA) is used extensively in BINARY and its subroutines. All data tables searched in this way involve composition as the independent variable. It has been found that a good comparison between computer and manual solution is achieved with a minimum of twenty data points (composition increment 0,05 mole fraction). The user may increase the size of the arrays, by changing a parameter in the COMMON block BINCOM, to accommodate a greater number of data for greater interpolation accuracy. For more than fifty points, however, the improvement in accuracy for ideal systems will not necessarily justify the increased core storage requirement. (See Chapter 8.)

Enthalpy data may be supplied in any convenient set of units. Conversion factors relating the data to the internally used British FPS system must also be supplied. Results are converted to the SI before printing to conform to the metrication regulations. This procedure has been adopted because many of the data and correlations quoted in the literature are in the FPS system. The use of conversion factors by the program is discussed fully in the User Manual.

Error messages result if material balances fail or if a pinch point is detected. Errors detected in subroutines causes MALF=1000 and this, on return to the mainline, causes termination of the run. (See User Manual, Chapter 7.)

This program will enable users to investigate the effects of parameters such as reflux ratio and stage efficiency. The relative merits of the two design procedures can also be compared.

3.5 SUBROUTINES ASSOCIATED WITH MAINLINE BINARY

Only those subroutines not discussed in Chapter 2 appear here.

3.51 Subroutine READBD

This subroutine is used to read all the data required to construct either McCabe-Thiele or Ponchon-Savarit diagrams for binary continuous distillation systems. It is used both by BINARY and by PACKED, which uses a Ponchon-Savarit diagram in the design of packed columns. Counter NOP defines the design procedure required by the user.

NOP=1 Tray column McCable-Thiele diagram.
NOP=2 Packed column Ponchon-Savarit diagram
NOP=3 Tray column Ponchon-Savarit diagram

In addition to accepting data READBD computes equilibrium data as described in section 2.21 and calls subroutine FLASHB to determine the feed condition for Ponchon-Savarit designs. A sample runstream showing the conversation data-input system is given in sets MCAN=2 which suppresses the conversational output. The output formats in the text serve to document the routine. Intermediate calculations are described below. Details of the data required and the input formats are given in the User Manual (section 7.2).

Line No.	Program Action
15-19	Conversion of supplied pressure units to mm Hg. The system pressure is required for computing bubble points and dew points using Antoine's equation. Antoine constants are usually related to vapour pressure in mm Hg and hence it is convenient to convert the total pressure to the same units.
29-98	Calculation of vapour-liquid equilibrium data as described in Chapter 2. Values are stored in arrays XEQUIL and YEQUIL for use by plotting routines.
37-54	Calculation of x-y and α data using Antoine constants (Equation (2.6) and (2.2)).
67-74	Calculation of x-y data using a constant relative volatility. (Equation (2.2)). This option is not available to Ponchon-Savarit designs.
(85-98)	Calculation of α data using supplied x-y data (Equation (2.2)).

Note: The calculation or specification of equilibrium temperature data, TALF, is required for Ponchon-Savarit designs. If x-y data are supplied for a McCabe-Thiele design, array TALF may be filled with zeros (see User Manual).

113	Subroutine CONDEN is called to determine the composition and phase distribution of the distillate product. (See section 3.52.)
114	Complex columns may be specified only in McCabe-Thiele designs.
173-177	Correcting the enthalpy function coefficients. (See User Manual.)
185	Subroutine FLASHB called to determine the feed condition. (section 3.53)

3.52 Subroutine CONDEN

This subroutine determines the composition of the product stream leaving the condenser. Either a partial or a total condenser may be specified. When CONDEN is called the distillate product composition XD and the type of condenser are known.

Line No.	Program Action
(a) Total condenser (NCOND=1)	
7-9	XD is the composition of the liquid distillate product. XC and YC are set equal to XD and control is returned to the calling routine. YC is the composition of the vapour in the condenser (see Figure 3.2 (a & b)) and is not equal to XD for total condensers. In the program, however, YC is not required as a composition, but only as value for terminating the stepping procedure in subroutine STEPD (see section 3.516, line 21). Thus setting YC=XD here is convenient and does not cause any errors in the final results.
(b) Partial condenser (NCOND=2)	
12-15	Routine determines the fraction of each phase in the total distillate product stream.
(i) Single phase distillate product (all vapour) (Figure 3.2(a))	
17-21	For vapour distillate, YC=YD and XC is determined from equilibrium data using XYALFA. This is the only partial condenser type allowed by the program in Ponchon-Savarit designs.
(ii) Two phase distillate product (Figure 3.2(b))	
23-31	Routine determines the significance of the specified composition XD.

32-39 For specified XD referring to vapour phase, i.e. YC=XD, the corresponding liquid phase composition XC is determined using equilibrium data. The true overall composition is then computed as a new XD.

41-49 For specified XD referring to liquid phase the corresponding vapour phase and overall compositions are computed similarly.

50-63 If XD is specified as the overall composition, phase compositions YC and XC are determined using the standard iterative search procedure . (See subroutine SEARCH, section 2.5).

3.53 Subroutine FLASHB

The feed condition is determined by this subroutine using the equations described in section 3.2. A flash calculation is unnecessary when considering ideal binary systems for which equilibrium temperature composition data are known.

Line No.	Program Action
5-7	Compute bubble and dew points of feed and test to determine whether or not it is superheated or supercooled.
(a) Two phase feed	
8	Call subroutine EQUIL to determine equilibrium vapour-liquid composition of feed based on known feed temperature.
9	Compute feed condition, Q (Equation 3.27 with x_F, y_F substituted as x_q, y_q).
(b) Superheated or supercooled feed	
12-15	Determine enthalpy of feed at specified temperature (HVF or HLF) and at a saturated vapour (HV) and liquid (HL).

16	Compute feed condition Q (Equation 3.26).
18-21	Feed q-value and temperature data are printed out for inspection.

3.54 Subroutine EQUIL

The standard linear interpolation search technique described in section 2.53 (subroutine XYALFA) is used here to determine the equilibrium liquid and vapour compositions of a two phase feed at the specified feed temperature, TF. Equilibrium temperature composition data stored in TALF, XALF and YALF are searched for the required values. The selected compositions are returned through the calling list as X1 and Y1.

3.55 Subroutine MULTFS

This subroutine is called by BINARY for McCabe-Thiele design of complex columns. When control is returned to the mainline, the x,y diagram is ready for stepping off equilibrium stages.

Line No.	Program Action
3-34	Overall material balance calculation. In the event of failure an error message is printed out and the program is terminated. (Equation (3.6))
37-38	Coordinates of point of intersection of overall operating lines (Equation (3.9)).
39-44	Slope of overall bottom operating line is then calculated and hence the flowrates in the bottom section of the column.
46-78	Feed and side streams may be specified in any order by the user. 'DO' loop 21 arranges the streams in order of increasing composition of the light component. They are then positioned

in the column in this order to minimise the number of trays required. The effect of inserting a more concentrated feed below a less concentrated feed may be investigated manually by the student, but is not considered in the program.

79-85 Bottom operating line equation slope and intercept stored.

86-94 'DO' loop 24 computes the points of intersection, slopes and intercepts of the operating lines defined by the order of the feed and side streams. The flowrates in each section of the column are also calculated.

98-107 Error messages are printed out if the material balance fails or if two streams with the same composition are found. MALF is set to 1000 to terminate the mainline on return.

3.56 Subroutine SIMPLE

This subroutine is called by BINARY for McCabe-Thiele design of simple columns.

Line No.	Program Action
6,7	Point of intersection of operating lines for a simple column is computed as XQM,YQM (Equation (3.9)).
9-19	Calculation of bottoms flowrate and composition by material balance for either partial reboiler or open steam operation (Equation (3.6)).
11 and 19	Material balance failure checks.
22-33	Development of operating line equations and calculation of vapour and liquid flowrates in each section of column.

34-39 Points of intersection of operating lines with the diagonal and with each other are stored for use in subroutine STEPD.

42-46 Error message is printed out and program is terminated if the material balance fails. MALF=1000 causes termination on return to main-line.

3.57 Subroutine PONSAV

This subroutine constructs a Ponchon-Savarit operating diagram. The enthalpy concentration diagram is set up first and the operating lines on the x,y diagram are defined from this. Condenser and reboiler heat duties are computed. The subroutine is called by BINARY and by PACKED (see Chapter 4).

Line No.	Program Action
3	Heat duty and stream enthalpy conversion factors.
4-6	Calculation of bottoms flowrate and composition by material balance, with check for failure. Only partial reboiler operation is considered (Equation (3.6)).
13-29	'DO' loop 10 computes data required for construction of enthalpy-concentration diagram. Procedure as follows.
(14-23)	(i) At vapour and liquid composition increments of 0,05 mole fraction, dew point and bubble point are calculated in subroutines BINBPT and BINDPT respectively (see section 2.52).
(24)	(ii) Subroutine HTCAP computes vapour and liquid enthalpies for the current composition and temperature data. (See section 3.58.)
(25,26)	(iii) Enthalpy data converted to SI units.

31-36	Feed enthalpy calculated similarly using HTCAP.
37	Subroutine RMIN computes the minimum reflux ratio (section 3.511).
39	This value is compared with the specified reflux ratio.
41	If the specified value is too low a value of 1,5 times the minimum is assumed. The user is informed of this action by error message. See line 106 below.
43,44	Distillate product enthalpy determined by either total or partial condenser specification. (H_D on Figure 3.3.)
45-51	Development of 'Δ' points on enthalpy-concentration diagram (Equation (3.21-3.23)).
52,53	The equation of the line passing through these points and the feed enthalpy is defined. (Line Δ_B-F-Δ_D.)
57-77	Search for point of intersection of the operating lines, which in general are not straight. Subroutine SEARCH is used in the iterative search for the correct composition values as described in Chapter 2.
78	Subroutine OPLINE computes the coordinates of a set of points defining each operating line. (See section 3.512.)
80	Subroutine FLOWS calculates the vapour and liquid flowrates at each point in the column defined by OPLINE. (See section 3.513.)
82,83	Conversion of heat units from FPS system to the SI.

85-97 The points of intersection of the operating lines with the diagonal and each other, are stored for use in the stepping procedure. These calculations are not required in packed column designs and are therefore avoided (NOP=2).

99-110 Error messages are printed out if the material balance fails or if the specified reflux ratio is less than the calculated minimum. The former error terminates the execution.

3.58 Subroutine HTCAP

This subroutine calculates vapour and liquid enthalpies at specified points through the column, thus defining the enthalpy-concentration diagram for Ponchon-Savarit designs. The calling list supplies the temperature and composition data and returns the calculated enthalpies to the calling program as HV and HL. The calculations are performed using the equations discussed previously.

Line No.	Program Action
5-7	Convert temperature values to units required by function coefficients (see User Manual).
8-14	'DO' loop 10 computes the molar enthalpy of each component in the vapour over the temperature range TD-T where T is some base temperature (integral of Equation (3.19)).
15-16	Calculation of enthalpy contribution of each component based on composition YS,(1-YS). Conversion factors CPMW and CPCONV ensure consistency of units (Equation (3.16)).
17-20	'DO' loop 11 computes latent heat of solution at vapour dew point (Equation(3.20)).
21	Total enthalpy of saturated vapour at temperature TD (Equation (3.18)).

22-28 'DO' loop 20 performs the same calculations as 'DO' loop 10, for the liquid phase. (Equation (3.17).)

29 Heat of solution may be included if data are available. Subroutine HTSOLN is called to calculate this value.

34 Total enthalpy of saturated liquid at temperature TB. (Equation (2.15).)

3.59 Subroutine HTSOLN

This subroutine is called by HTCAP to compute liquid phase heats of solution if data are supplied. It is assumed that the base temperature taken for the enthalpy calculations is the temperature at which heat of solution data are quoted.

The standard linear interpolation search procedure is used to determine a value H for a solution of composition XA. HEVABS is equal to either +1 or -1 for endothermic or exothermic mixing respectively.

3.510 Subroutine HFIND

The vapour and liquid enthalpy data calculated in HTCAP is stored in arrays HVAP and HLIQ resepctively, with corresponding composition arrays YH and XH. Subroutine HFIND accepts enthalpy-composition data for either phase through HENTH and XYH. The standard linear interpolation search procedure is used to determine the enthalpy H of a solution of composition XA using these data. NENTH data points are contained in the data arrays. The search procedure is discussed in subroutine XYALFA, section 2.53.

The dimension statement expands automatically to accommodate the particular arrays carried through in the calling list. This reduces the total core requirements during execution. (See Chapter 8.)

3.511 Subroutine RMIN

This subroutine calculates the minimum reflux ratio allowed by the specifications for a Ponchon-Savarit design problem. The routine is called by subroutine PONSAV as part of the enthalpy-concentration diagram construction. The procedure followed is discussed in section 3.22. The programming of this procedure is described below.

The standard iterative search technique, using subroutine SEARCH, is used to determine the tie line passing through the feed point on the diagram. The steps are as follows. (Refer to Figure 3.3.)

Line No.	Program Action
5	Initial vapour concentration chosen Y1=XD.
11,12	XYALFA called to determine relative volatility ALF corresponding to Y1 and equilibrium liquid composition X1 is calculated.
16,17	HFIND uses linear interpolation search procedure to determine enthalpies of vapour and liquid of composition Y1 and X1.
20,21	Test if feed point (enthalpy HF) lies on the tie line.
23	SEARCH called for new estimate of Y1.
24	Return to repeat lines 11-23 above.
25	When HF falls on line within limit ACC (line 3), search is complete. Point X1,Y1 is the point of intersection of the two operating lines and the equilibrium curve.
28	Determine enthalpy of reflux liquid using HFIND. For a total condenser this is equal to the distillate product enthalpy (XC=XD).

30	Determine enthalpy of top vapour for a total condenser operation.
32-46	Determine enthalpy of top vapour for a partial condenser operation.
48	Compute minimum reflux ratio RM. (Equation (3.25))
50-71	Error messages prevent excessive looping in search procedures, all of which cause termination of the run.

3.512 Subroutine OPLINE

This subroutine uses the standard Ponchon-Savarit graphical projection technique to determine the coordinates of points on the x,y diagram defining the operating lines. OPLINE is called after the relevant data on the enthalpy-concentration diagram has been computed. The projection technique is programmed as follows (refer to Figure 3.3):

Line No.	Program Action
7	Divide interval (XD-XCR) into NINT(50) increments of size DELX.
11-35	For each value of X1 between these limits determine the ordinate YG defining a point on the operating line as follows. [This ordinate is given as the point of intersection of the vapour-enthalpy curve and the external reflux line passing through X1 (line X1ΔD on Figure 3.3).]
(15)	Use HFIND to determine enthalpy of liquid X1.
(16-17)	Define equation of external reflux line passing through X1.
(23-31)	Use iterative search procedure SEARCH to find point of intersection of vapour enthalpy curve and reflux line, hence YG. Repeat 11-35 for each X1.

33,34 Values are stored in arrays XROP, YROP.

37-72 Above procedure repeated for each point defined in the stripping section interval (XCR-XB). Coordinates are stored in XSOP, YSOP. The increment size in this section is the same as in the rectifying section (see below). Lines 37-40 determine the number of increments and check this against the dimension parameter NPPSC.
Note: NPPSC=NPPS-2, NPPS defined in PARAMETER statement.

73-83 Error messages prevent excessive looping in search procedures. This could be avoided by increasing the error limit ACC (line 3) which is presently set at 0,01%.

The data computed here is used by PACKED as well as BINARY. PACKED calculates mass transfer coefficients and uses a Simpson's Rule numerical integration technique to calculate the height of packing required in each section of the column. The size of increment DELX above is thus kept constant throughout the column to maintain a consistent level of accuracy. It is possible that the size of the interval XCR-XB could be large enough relative to (XD-XCR) to cause the number of intervals to exceed the dimensions of XSOP and YSOP. Dimension parameter NPPS could be increased to avoid this if necessary. A situation such as this is unlikely, however, as it would require a high feed concentration, or a highly supercooled liquid, or a combination of both to cause it. The use of NINT=50 with NPPS=150 proved to be sufficient for all examples attempted, and results agreed well with manual solutions (see Chapter 8).

3.513 Subroutine FLOWS

This subroutine determines the vapour and liquid flowrates at each point in the column defined by OPLINE. Standard mass balance equations are used and both mass and molar flowrates are computed.

The mass flowrates are required in the design of packed columns only. Subroutine AVEMW is called to calculate solution molecular weights for both phases. The bubble point or dew point of each stream is also computed.

Line No.	Program Action
9-17	Calculation of data for reflux liquid and top tray vapour streams.
18-34	'DO' loop 11 calculates the data corresponding to all streams in the rectifying section of the column.
35-50	'DO' loop 12 duplicates 'DO' loop 11 for the stripping section.
51-52	Reboiler stage is stepped off in the normal way to determine the degree of separation achieved. For some systems such as Acetone-Water, this step indicates that the feed should be entered below the first tray, which means that the column has no stripping section.
54-57	Message printed out to inform the user if no stripping section is considered.
58 and 67	At this point the calculations required in tray column designs are complete and control is returned to the calling routine.
59-65, 75-89	For packed columns the reboiler step sets the conditions at the bottom of the packing. The relevant data for packed columns that have no stripping section are calculated here.

68-74, 90-103	Conditions at bottom of the stripping section in a packed column are calculated here.
105-123	Error messages indicate detected inconsistencies such as negative flowrates.

3.514 Subroutine AVEMW

This subroutine calculates the molecular weight of a vapour or liquid mixture of composition Y1 or X1 respectively. It is used by Subroutine FLOWS to determine mass flowrates, through the column for use in packed column designs. (See Chapter 4.)

3.515 Subroutine CHEC

This subroutine checks that no pinch points occur on the x,y diagram before an attempt is made to step off the equilibrium stages. This is done by checking that all points of intersection of pairs of operating lines lie below the equilibrium curve. If a pinch point is detected an error message is printed out and the execution is terminated.

Line No.	Program Action
5	Selection of abscissa of operating line intersection points, XA.
6-7	Calculation of equilibrium vapour composition corresponding to XA.
8	Check to ensure that the point of intersection does not lie either on or above the equilibrium curve.
10-17	Error message indicates position of any pinch point found. Error results in termination of execution.

3.516 Subroutine STEPD

This subroutine steps off equilibrium stages on an x-y diagram for either McCabe-Thiele or Ponchon-Savarit designs. A different procedure is used for each design method with counter NOP selecting the appropriate calculations. A Murphree tray efficiency factor is included and the change from one operating line to the next is done so as to minimise the total number of stages.

Line No.	Program Action
5-9	Stepping procedure is started from the bottom of the column. Starting position is set at XB,YB on the bottom operating line.
11	'IF' statement causes change from one operating line to the next.
14	Check to ensure that array dimension size is not exceeded.
15-18	Determination of point on equilibrium curve above point on operating line. (Equation (2.2) - vertical line of step.)
19	Adjustment of step size using tray efficiency factor (Equation (3.7)). The reboiler is assumed to be a perfect stage and hence MT = 1 is not included.
20	Same as line 11. Check within inner 'DO' loop 11.
21	Procedure terminated when the point of intersection of the top operating line and the diagonal is reached. YC set in subroutine CONDEN.
22-23	Determination of operating line abscissa corresponding to point found by lines 15-19 (horizontal line of step), for McCabe-Thiele design (Equation 3.3 as written for each section of column).

24-26	Subroutine PSOPXY is called to find abscissa for Ponchon-Savarit design.
28	'DO' loop 11 returns to repeat calculations to define each stage in the column.
31	Determination of starting point on following operating line after a change is indicated by line 11 or 20 above, for McCabe-Thiele designs. (Equation (3.3)).
34,35	Same calculation for Ponchon Savarit design using PSOPXY.
37-41	Storing of feed and side stream tray positions defined as those trays straddling points of intersection of the appropriate operating lines.
44-45	Calculation of the fraction of a theoretical stage required to complete an ideal column. (Equation (3.10).) TRAYNO is the total number of theoretical stages required.
48-68	Determination of vapour and liquid flows leaving each tray in a Ponchon-Savarit design. Subroutine FFIND uses the standard linear interpolation search procedure to extract data from arrays set up by FLOWS. (See note below.)
70-77	Error message. If the number of stages stepped off exceeds the array dimension limits the procedure is terminated and operating diagram data is printed out. This will reveal the types of pinch point caused by deviations from Raoult's law. If a pinch point is not indicated, the array size should be increased.

Note: Arrays LRWT, VRWT and TRV are used to store flow and temperature data for Ponchon-Savarit designs of tray columns. Their definition in the nomenclature applies only to their

application in packed column designs. This is done to reduce the total number of dimensioned arrays.

3.517 Subroutine PSOPXY

This subroutine is called by STEPD in Ponchon-Savarit designs. Its purpose is to return the abscissa of a point on an operating line corresponding to the ordinate Yl, which has been set by a point on the equilibrium curve. This determines the horizontal line of the stepped stage on the x,y diagram.

The routine uses the standard linear interpolation search procedure to extract a value from either XROP or XSOP. These arrays contain x-coordinates of points on the top and bottom operating lines respectively previously determined by subroutine OPLINE.

Line No.	Program Action
3-15	Calculations for points lying on the bottom operating line.
17-29	Calculation for points lying on the top operating line.

3.518 Subroutine FFIND

This subroutine is called by STEPD to determine the molar vapour and liquid flows leaving each stage, and the temperature on each stage, defined on a Ponchon-Savarit x,y diagram. The linear interpolation search procedure is used. A COMMON block is not included and the data arrays are provided through the calling list. The internal arrays FVAP, FLIQ, FTEMP and FX are expanded to accommodate the data arrays VAPR, LIQR, TRL and XROP or VAPS, LIQS, TSL and XSOP respectively, depending on the section of column referred to. These data arrays are compiled in FLOWS.

Using a calling list and dimension statement in this way reduces the total core requirements for running the program. (See Chapter 8.)

3.519 Subroutine BINRES

This subroutine prints out the results of either a McCabe-Thiele or a Ponchon-Savarit design of a tray column. The format statements are self-explanatory and counter NOP selects the appropriate commands. Tables of data describing the equilibrium curve, operating lines and equilibrium stage positions may be printed out on request. These data could then be used to construct an operating diagram, or to check a previously constructed diagram. The tables are lengthy and it is thus often convenient to suppress them. The data in the tables is available on file should the user wish to plot a diagram on the CALCOMP plotter (see User Manual). Sample printouts are shown in Chapter 8.

3.6 SUMMARY OF BINARY CONTINUOUS DISTILLATION PROGRAM

Program name: BINARY
Type: Mainline with subroutines.
Operation mode: Demand or Batch on UNIVAC 1106 - EXEC 8 system.

1.Title: Design of binary continuous distillation tray towers by methods of McCabe and Thiele, and Ponchon and Savarit.

2. Solution type: Design program.

3. Application: BINARY can be used to design binary continuous distillation systems by McCabe-Thiele or Ponchon-Savarit procedures.

a. McCabe-Thiele:
The program is limited to the following,

1) Standard McCabe-Thiele assumptions.
2) Thermodynamically ideal systems.
3) Simple or complex columns.
4) Partial or total condenser operation.
5) Partial reboiler or open steam operation.
6) Single Murphree tray efficiency factor for all trays.
7) Assumed zero pressure drop across trays.

b. Ponchon-Savarit:
The program is limited to the following,

1) (2), (4), (6) and (7) above.
2) Simple columns (one feed, no side streams).
3) Partial reboiler operation.

4. Program Development:

a) Computer system UNIVAC 1106 - EXEC 8.
b) Minimum core requirements: 14,5 K
c) Language: FORTRAN IV with modifications permitted on above system.
d) Typical run time: McCabe-Thiele: 0,6 secs
Ponchon-Savarit: 4,6 secs

CHAPTER 4

BINARY CONTINUOUS DISTILLATION - PACKED COLUMN DESIGN

4.1 INTRODUCTION

Although most industrial rectifying columns are tray devices, packed columns are better suited for the following applications [58]:

(1) Handling of corrosive fluids;

(2) Food and pharmaceuticals, where contact with metal could cause harmful contamination;

(3) Low pressure distillation, which requires a low pressure drop;

(4) Distillation of heat sensitive materials, where high vapour-low liquid flowrates are maintained;

(5) Small-scale application, where small cross-sectional area requirements reduce the efficiency and increase the relative cost of tray towers, e.g. pilot plant and laboratory studies.

Peters and Timmerhaus [59] and King [60] tabulate a number of points to be considered in selecting either a tray or a packed column for a given service. These confirm the suitability of packed columns for the applications listed above. In particular:

(a) Random-packed columns seldom exceed 1,2 metres in diameter while the diameters of industrial tray columns are usually greater than 0,6 m.

(b) Packed columns are cheaper and easier to construct in the following instances:

i) Corrosive fluid handling

ii) Vacuum operation - the packing reduces the

possibility of tower wall collapse and in general, provides a lower pressure drop than a tray column designed for the same duty.

(c) Liquid hold-up is considerably less in packed columns.

4.2 THEORY OF BINARY DISTILLATION IN PACKED COLUMNS

4.21 General Design procedure

In packed columns there are no discrete identifiable points at which the vapour and liquid streams are considered to be in equilibrium. Thus, rather than the stepwise change in concentration with height achieved in tray towers, the concentration changes in packed columns are continuous and differential. This differential change must be accounted for in the computational procedure.

Figure 4.1 shows a schematic diagram of a packed column, a corresponding operating diagram, and a differential section of packing. Lengths Z_R and Z_S represent sections of packing (see Figure 4.1 (a)), contained between support plates. Plug flow is assumed for each phase, thus implying no axial mixing. This permits the use of standard material and energy balance equations. Operating lines may therefore be constructed on an x,y diagram, as for a tray column design. The procedure for stepping off stages does not apply in packed column design however.

The height of packing required must be determined from a consideration of the rate of mass transfer of material across the vapour-liquid interface.

The two primary mechanisms of mass transfer considered are molecular diffusion and eddy diffusion. Molecular diffusion is the dominant mechanism under laminar flow conditions whereas eddy diffusion is rate controlling in the

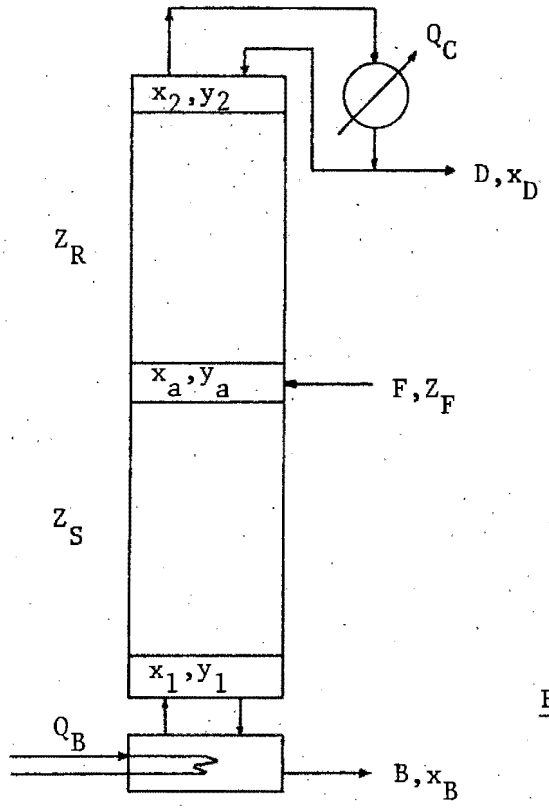


FIGURE 4.1(a) PACKED COLUMN

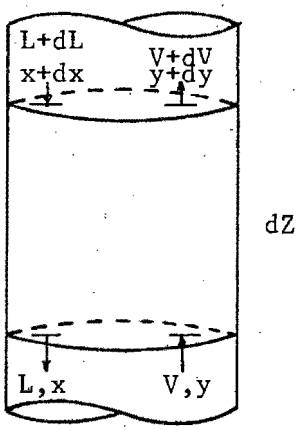


FIGURE 4.1(c) DIFFERENTIAL HEIGHT OF PACKING

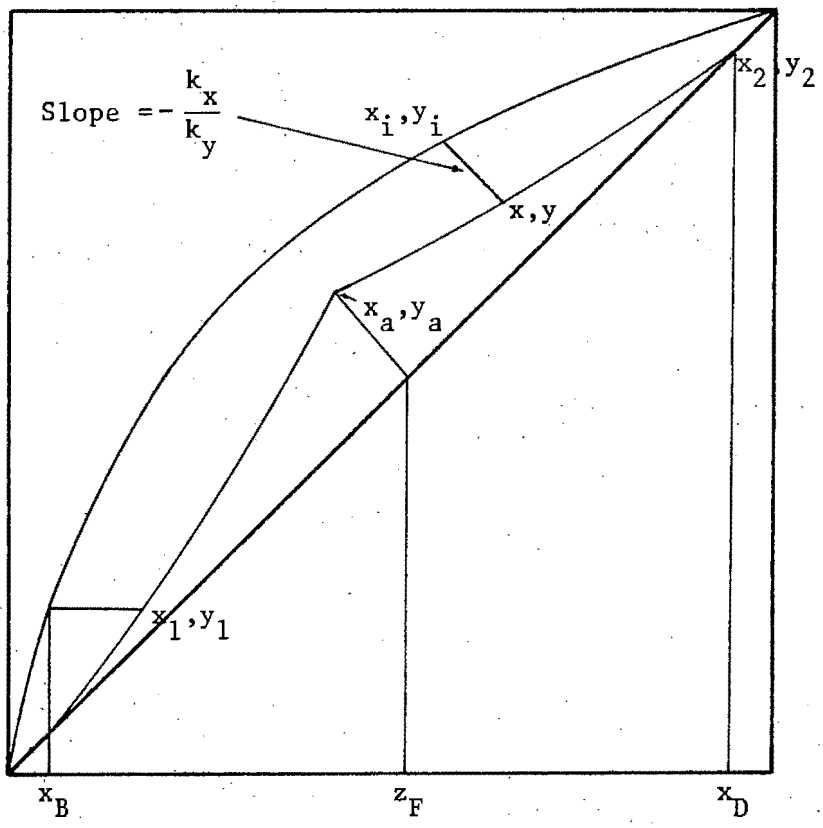


FIGURE 4.1(c) x,y DIAGRAM - PACKED DISTILLATION COLUMN

turbulent regime. In between the two flow regimes both mechanisms contribute to mass transfer.

In this study the vapour-liquid equilibrium curve and the operating lines are determined as in a Ponchon-Savarit design of a tray column (see Chapter 3). Mass transfer coefficients are calculated directly using the empirical relationship proposed by Shulman et al. [61]. The height of packing required is computed using the procedure suggested by Treybal [9]. A numerical integration technique is used instead of Treybal's graphical procedure.

4.22 Mass transfer analysis for packed column design

The theoretical analysis of mass transfer between phases is discussed in several of the texts referred to in the undergraduate course [9,58,62]. A discussion of the theory relating to the design of packed distillation columns is given here.

The rate of mass transfer across an interface is proportional to the difference between the bulk phase concentration and the concentration at the interface. The constant of proportionality is the phase film mass transfer coefficient. This rate of transfer can be expressed as follows:

$$N_A = k_x(x-x_i) = k_y(y_i-y) \qquad (4.1)$$

The bulk phase concentrations are related by the operating line equation for the system (Equation 3.3). For a counter-current column Equation (4.1) can thus be solved graphically, as shown in Figure 4.1 (c), or analytically, by simultaneous solution of Equation (4.1) and the equilibrium relationship, Equation (2.2), provided that the ratios of the film coefficients k_x and k_y can be determined for sets of values of x and y.

The shortest length of packing in each section required to achieve the given separation at a given reflux ratio, is defined when the bulk phase compositions at the feed point are x_a and y_a (Figure 4.1 (c)). This corresponds to the stepping procedure resulting in the least number of stages in a tray column.

In packed columns, flowrates are based on unit column cross-sectional area for use in flooding and mass transfer coefficient correlations. The height of packing required in each section is determined as follows.

Consider the differential height element, dZ, in Figure 4.1 (b). The interface available for mass transfer within this section is given by adZ, where a is the specific interfacial surface area for distillation (see section 4.343 (c)).

The amount of component A passing through the element of height dZ is $V.y_A$ moles A/hr m^2 in the vapour phase and $L.x_A$ moles A/hr m^2 in the liquid phase.

The rate of mass transfer is thus

$$N_A = \frac{dVy}{adZ} = \frac{dLx}{adZ} \qquad (4.2)$$

Equation (1.5) is derived by combining Equation (4.1) and (4.2), and rearranging; thus for the rectifying section

$$Z_R = \int_0^{Z_R} dZ = \int_{Vy_a}^{Vy_2} \frac{dVy}{k_y a (y_i - y)}$$

$$= \int_{Lx_a}^{Lx_2} \frac{dLx}{k_x a (x - x_i)} \qquad (4.3)$$

A similar expression, with appropriate limits, may be written for the stripping section.

Equation 4.3 applies to the general case, but does not lend itself to manual calculation procedures. Most manual procedures reviewed in the literature involve at least one of two standard simplifications [17,58,62]. These are:

(a) McCabe-Thiele assumptions:

Assuming constant molal overflow, the operating lines are straight and the concept of a Height of Transfer Unit (HTU), introduced by Chilton and Colburn [58], may be used. For a distillation process the HTU is related to the mass transfer coefficient by the equation

$$\text{Liquid phase} \quad H_{tL} = \frac{L}{k_x a}$$

$$\text{Vapour phase} \quad H_{tV} = \frac{V}{k_y a} \qquad (4.4)$$

Hence Equation (4.3) may be written as

$$Z_R = H_{tV} \int_{y_a}^{y_2} \frac{dy}{y_i - y} = H_{tV}\, N_{tV} \qquad (4.5)$$

$$\text{or} \quad = H_{tL} \int_{x_a}^{x_2} \frac{dx}{x - x_i} = H_{tL}\, N_{tL} \qquad (4.5)$$

A number of empirical relationships are available for calculating HTU's [58]. As mass transfer coefficients and interfacial area depend on mass flowrates rather than molar flowrates, McCabe-Thiele assumptions can not ensure that the HTU's are constant. This aspect must thus be checked before using the procedure.

(b) If, in addition to straight operating lines, the equilibrium curve can be considered as a straight line,

(4.6)

Equation (4.5) can be written in terms of "overall" values.

$$Z_R = H_{tOV} \int_{y_a}^{y_2} \frac{dy}{y^*-y} = H_{tOV}\, N_{tOV} \qquad (4.6)$$

In such instances the number of transfer units may be determined without graphical integration, using empirical relationships. This method has limited application but is useful for computing a rough estimate manually.

The program developed here uses the general expression shown in Equation (4.3). The above manual procedures are used by students for comparative purposes.

Considering Equation (4.3), either integral may be used to determine the height of packing required. The choice depends on the relative magnitudes of k_x and k_y.

For $k_x > k_y$ the principal resistance to mass transfer lies in the vapour phase. In Figure 4.1 (c) (y_i-y) is greater than $(x-x_i)$ and hence can be determined more accurately. Therefore the expression involving (y_i-y) in Equation (4.3) would be used in the integration. Similarly for $k_x < k_y$ the expression in terms of $(x-x_i)$ would be used.

In general distillation processes are vapour-phase controlled with $k_x > k_y$ [63].

4.23 Design Calculations

4.231 Calculation procedure: The sequence of design calculations for a packed column used in this study runs as follows:

(1) Determine operating diagram for system using standard Ponchon-Savarit procedure. The reboiler, considered to be an ideal equilibrium stage, is stepped off

as shown in Figure 4.1 (c), thus defining point (x_1, y_1).

(2) Choose a number of equally spaced points on each operating line from (x_1, y_1) to $(x_a y_a)$ and from $(x_a y_a)$ to (x_2, y_2) (see Figure 4.1 (c)). The greater the number of points, the more accurate will be the numerical integration of Equation (4.3). The interval between points is the same for both operating lines to ensure similar accuracy in the integration procedure for each section.

(3) At each chosen point determine the vapour and liquid mass and molar flowrates, temperatures and compositions.

(4) Select the point in the column at which the liquid flow is greatest. It is assumed that if the column is designed so as not to flood under the conditions of flow at this point, it will not flood anywhere else [9].

(5) A pressure drop is chosen such that the tower will operate at about 60% of the flooding velocity. This value is usually 0,75-1,0 in. H_2O/ft packed height for distillation at a total pressure of 1 atm and about 0,01-0,05 for vacuum distillation [17]. The generalized flooding and pressure drop correlation chart, shown in Treybal [9], see Appendix A, Figure 1, is used in the program.

Operation at 60% of the flooding value is usually adopted to ensure that the system will not become unstable [9,64], although some workers have reported a technique for operating a packed column under flooding conditions [65].

(6) The x-axis coordinate for the generalized flooding correlation chart is computed and the y-axis value is determined, corresponding to the chosen pressure drop curve. From this ordinate, the superficial vapour velocity at loading V' is calculated.

(7) The diameter of the column is then computed such that the vapour flowrate, at the selected point in the column, is less than or equal to the value at loading.

Alternatively, steps (5) to (7) are reversed. If a diameter is specified, the abscissa and ordinate values for the flooding correlation chart are computed and the resultant pressure drop is checked to ensure loading conditions. If necessary the diameter is adjusted and the calculations repeated until acceptable conditions are achieved.

(8) The height of packing required in each section is computed by numerical integration of Equation (4.3). The unknowns in the expression are determined as follows, for $k_x > k_y$ (see Figure 4.1 (c)).

At each chosen point on the operating lines, V and y in this equation are known. k_y, a and y_i must be evaluated. Composition y_i is found graphically by constructing a line of slope $-k_x/k_y$ from the known point (x,y) to intersect the equilibrium curve at (x_i,y_i). y_i is then read off the y-axis. The values of k_x,k_y and a are determined using Shulman's correlations discussed below.

This completes the design of the packed column as defined in this study. The diameter of the column and the height of packing in each section are calculated. The number and position of redistribution devices within the column are not considered. The diameter of the column is limited to 2 metres in the program as industrial columns seldom exceed this size [9].

4.232 Mass Transfer and Flooding Correlations: The mass transfer coefficient expressions used in this study have been extracted from the original work of Shulman as it appears in Treybal [9]. Treybal shows how the expression should be adjusted for distillation applications. Shulman's

analysis applies to Raschig rings and Berl saddles, both of which are common packing materials.

(a) Liquid phase mass transfer coefficient

In the liquid phase, Shulman et al. [61] proposed that the values of k_x and a be treated separately. It was proposed that the interfacial area should be obtained from available packing data and that the mass transfer coefficient should be determined using the following relationship

$$\frac{k_L d_S}{D_L} = 25{,}1 \left(\frac{d_S L'}{\mu''_L}\right)^{0{,}45} (Sc)_L^{0{,}5} \qquad (4.7)$$

For distillation the flowrates within any section of a column are usually sufficiently constant to permit the assumption of equimolar counter diffusion between the phases. Thus the local mass transfer coefficient at the phase interface, F, is equal to the film coefficients.

$$F_V = k_y \; ; \; F_L = k_x \qquad (4.8)$$

The film mass transfer coefficient k_L in Shulman's equation (4.7) was derived for the system, A diffusing into non-diffusing B. The conversion of k_L to F_L for such a system is given by

$$F_L = k_L \, x_{LM} \, c \qquad (4.9)$$

where x_{LM} is the log mean concentration driving force across the interface.

For the very low solute concentrations used in the derivation, x_{LM} approaches 1 and the following relationship holds

$$k_x = F_L = k_L \, c \qquad (4.10)$$

c is the molar density of the solvent liquid, or in the case of distillation, the heavy component.

(b) Vapour phase mass transfer coefficient

Shulman and Margolis [66] proposed the following expression for the vapour side mass transfer coefficient, based on the j-factor theory

$$F_V \frac{(Sc)_V^{2/3}}{V} = 1{,}195 \left[\frac{d_S V'}{\mu''_V (1-\varepsilon_0)} \right]^{-0,36} \qquad (4.11)$$

where $\varepsilon_0 = \varepsilon - \phi_t$ is the operating void volume in the packing. ε is the characteristic void volume of the dry packing and ϕ_t, the total liquid hold-up, is defined in part (c) below. The mass transfer coefficient F_V included in the relationship is the local phase-interface value, and from Equation (4.8) $k_y = F_V$.

(c) Interfacial area for mass transfer

The interfacial area for distillation, as derived by Shulman is given by the following expression [9]

$$a = a_{AW} \frac{\phi_0}{\phi_{0W}} \qquad (4.12)$$

where ϕ_0 and ϕ_{0W} account for the effect of liquid hold-up in the packing. In the original work of Shulman et al. [61], three types of liquid hold-up are defined, namely total, static and operating hold-up. These are related by the expression

$$\phi_t = \phi_s + \phi_0 \qquad (4.13)$$

Using water and air in columns containing various packings, Shulman derived empirical expressions for calculating the hold-up of water in a packing and hence the hold-up of a process liquid for which certain physical properties are known.

For water, or a very dilute aqueous solution

$$\phi_{tW} = \phi_{sW} + \phi_{0W} \tag{4.14}$$

Equations (4.13) and (4.14) are related by the expression

$$\phi_0 = \phi_{0W} H^* \tag{4.15}$$

where H* is a dimensionless correction factor. Empirical expressions defining ϕ_s, ϕ_{tW}, ϕ_{sW} and H* are shown in Appendix A, Table 1. These expressions were derived by Shulman [61], and are tabulated in Treybal [9].

The interfacial surface area for mass transfer in the air-water system, a_{AW}, was defined by Shulman as

$$a_{AW} = m(V')^n(L')^P \tag{4.16}$$

for columns operating under loading conditions. For systems with vapour phase densities similar to that of air Equation (4.15) may be used directly in Equation (4.13). For other vapour densities V' in Equation (4.15) is replaced by $V'.(0{,}075/\rho_V)^{0{,}5}$†. Values of the coefficient and exponents in Equation (4.16) are shown in Appendix A, Table 2.

Note: The Shulman correlations require data in a variety of units. Conversion factors are built into the program using a DATA statement. These have been chosen to convert the units most often associated with variables quoted in the literature to the form required by the correlations. These factors may be changed by the user if necessary (see User Manual).

(d) Flooding correlation chart

The generalised flooding and pressure drop correlation chart of the U.S. Stoneware Co. used in this study has been

---

† Units of ρ_V - lb/ft^3.

published by a number of authors [9,17,62,64]. This chart is used to determine the superficial vapour velocity for a specified pressure drop through the column or vice versa.

The x-axis coordinate is expressed as

$$\text{ABCIS} = \frac{L'}{V'} \left(\frac{\rho_V}{\rho_L}\right)^{0,5} \tag{4.17}$$

and the y axis coordinate as

$$\text{ORD} = \left[(V')^2 \, C_f \, \mu_L'^{0,2} \left(\frac{\rho_W}{\rho_L}\right)\right] / (g_c \, \rho_V \, \rho_L) \tag{4.18}$$

The chart may be used in one of two ways in the program.

(i) In the program, if a pressure drop is specified, the value of ABCIS is computed, a polynomial is fitted to the specified pressure drop correlation curve and the value of ORD is determined. The superficial vapour velocity V' is then determined using Equation (4.14).

(ii) If a diameter is specified, V', ABCIS and ORD are computed using the above expressions and the values are printed out. The user then determines whether the point (ABCIS,ORD) lies within the loading zone and if necessary adjusts the diameter. (See section 4.34.)

Characteristic constants for the packings considered in this study are shown in Appendix A, Table 3.

4.24 Choice of column packing

The performance of a packed column is dependent primarily upon the type of packing used, and the material from which it is made. The choice of a particular packing is normally based on cost, type of fluids to be handled and efficiency considerations.

A number of packings were extensively studied by Shulman et al. [61] and two of these, namely ceramic Raschig

rings and ceramic Berl saddles, were considered by Treybal [9]. These two packings are commonly used in industrial columns because of their relatively low cost. They are well documented in the literature and are thus useful for teaching purposes. For these reasons they are considered in this study.

Data for various sizes of these two packings are included in the program. The user is able to insert data for other packings if necessary. (See User Manual, section 7.3.)

4.3 DESCRIPTION OF PROGRAM PACKED

4.31 General Structure

The overall flowchart of the packed column design program is shown in Figure 4.2. The functions of the main-line routine are as follows.

(1) Start execution and read in data via READBD.

(2) Set up Ponchon-Savarit operating diagram using PONSAV.

(3) Perform all column-sizing and height-of-packing calculations.

Various subroutines are called through the program, to perform repetitive calculations and to determine physical property data. Integration to find the height of packing required in each section is performed by the subroutine SIMPS (see Chapter 2).

The operating diagram is determined as described in Chapter 3 for Ponchon-Savarit designs. Only the mass transfer calculation section will be explained in detail here.

4.32 Limitations

The same limitations concerning ideal-system behaviour, discussed in Chapters 2 and 3, apply to PACKED. There are

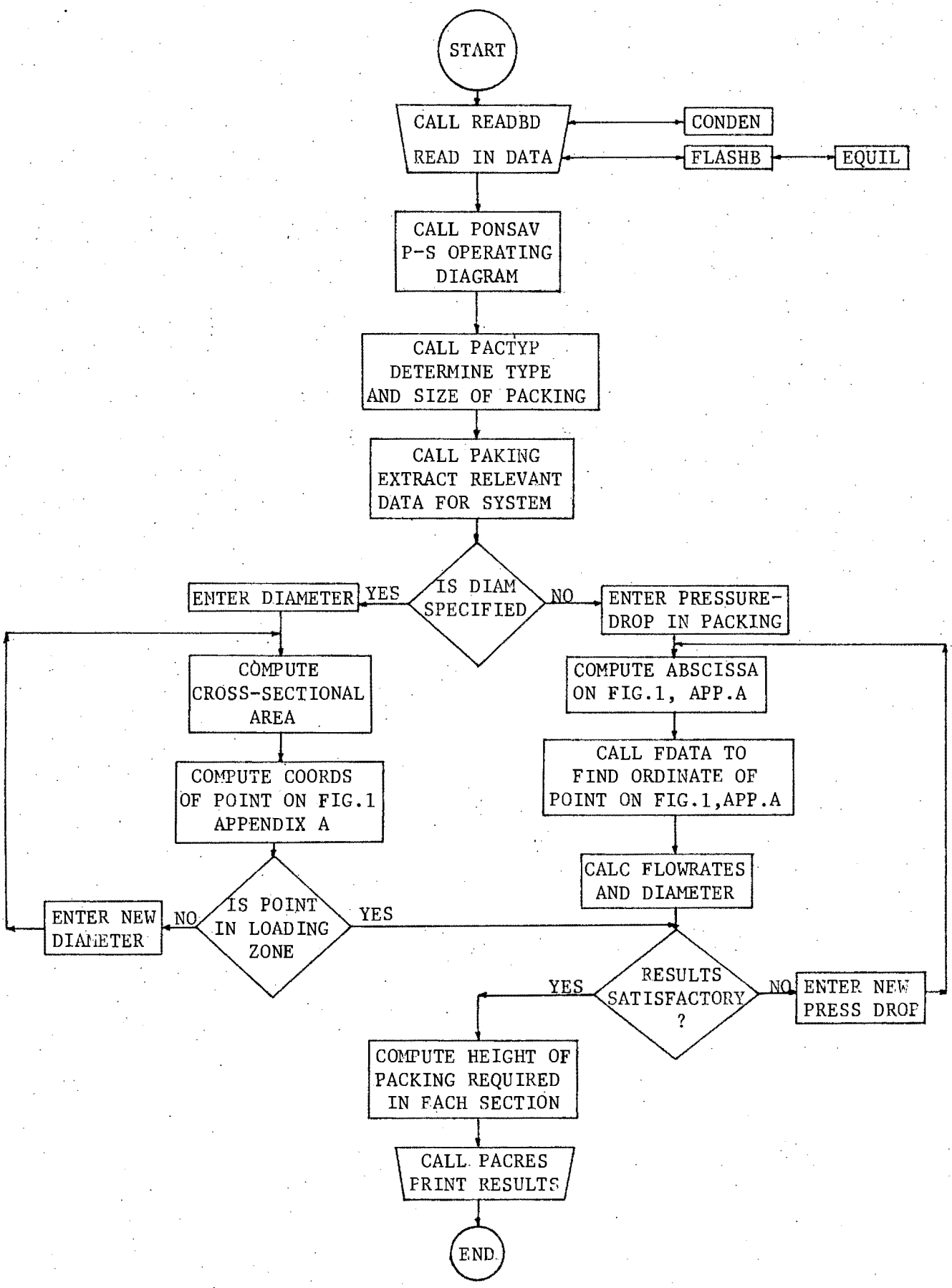


FIGURE 4.2 FLOWCHART OF PROGRAM PACKED

no plotting routines associated with PACKED and thus no data files need be assigned before the run. If a plot of the operating diagram is required, the problem should be solved using BINARY. This will also yield the number of theoretical trays equivalent to the designed height of packing. Data for four sizes of Raschig rings and three sizes of Berl saddles are included in the package. If other packings are specified the package data subroutine PAKING must be altered accordingly (see User Manual).

Internal conversion factors allow the user to supply data in any convenient system of units. Those factors currently built into the program apply to commonly encountered data tables. These are discussed individually in the User Manual and the user must ensure that the correct factor is available. Results are all returned in the SI.

4.33 Operating Procedure

It is recommended that this program be run only in demand mode. The program calculations are divided into two parts, as listed in section 4.31, and a separate data element may be added for each section. The user is required to check certain results visually, and to suggest changes if necessary, during the course of a run. In batch mode the outcome of these checks would have to be anticipated, which would defeat the purpose.

4.34 Program Calculations

Line No.*	Program Action
8,9	Conversion factors associated with units of physical property data and flowrates (see User Manual).
13-29	Start of program and reading of data through subroutine READBD as described in BINARY. The "first" data element contains all data pertaining to the operating diagram. (See User Manual.)

*See FORTRAN text, Appendix C.

33	Construction of a Ponchon-Savarit operating diagram as described in BINARY. On return to PACKED the reboiler stage has been stepped off (see subroutine FLOWS, section 3.513).
40-46	A second data element may be added to specify packing data. Alternatively data may be inserted conversationally.
47	Subroutine PACTYP determines the type and size of packing through conversational or data-element input.
49,50	Subroutine PAKING contains all the relevant data for the seven packings included in the program. Additional data may be inserted by the user if required. (See User Manual.)
51-58	The program computes solution-densities using pure component density data. If a table of solution-density data is available counter NRHOM is set from 0 to 1. (See section 4.551.)
60-66	The class into which each component falls is required for component density calculations. (See section 4.551.)
67-100	Array LRWT and LSWT, compiled by subroutine FLOWS, are examined to determine the point in the column with the greatest liquid flowrate. Once found, temperature and composition data at this point are noted and the point is identified by counter NPOINT. The column diameter will be designed such that the column will not flood at this point.
102-107	x-axis coordinate of generalized flooding and loading chart is calculated. Liquid density through subroutine RHOLIQ and vapour density

assuming ideal solution behaviour. (Equation (4.17).)

108-114 Optional specification of column diameter. If specified, the abscissa and ordinate on the flooding correlation chart are calculated. These values are pointed out for checking by the user to ensure that the point on the chart is within the loading zone. If not, a large diameter may be specified. (See section 4.232 (d) and the User Manual.)

117-121 Column cross-sectional area and superficial flowrates computed.

122 The calculation of ORD for pressure drop determination uses a higher superficial flowrate than the operating value. This represents a 5% safety factor for increases in operating vapour flowrate.

124-131 The alternative to specifying a diameter is to specify an allowable pressure drop across the packing at loading conditions.

135-137 VPDL is the pressure drop value displayed in the result table by subroutine PACRES. The choice of values corresponds to the curves on the chart, but the units have been changed to kPa/m to comply with metrication regulations.

138-142 Subroutine FDATA fits a polynomial to the selected curve and the resulting ordinate is printed out for checking by the user. The data fitted in FDATA is included in the program and was taken from the curves. The user may insert a set of data for a curve corresponding to a specific pressure drop if required. (See section 4.54.)

143-146 Viscosity and density calculations (values required in Equation (4.18).)

147-157 For specification of column diameter, ORD is computed (Equation (4.18)) and the coordinates of the point on the flooding correlation chart are printed out (see section 4.232 (d)(ii).) The user decides whether the specified diameter is satisfactory. If the point is in the flooding zone, a larger diameter should be specified.

159 Return to repeat calculations if necessary.

161,162 Determination of superficial vapour velocity from Equation (4.18) for specification of pressure drop (see section 4.232 (d)(i).)

163-177 Having selected a pressure drop, the column diameter is computed accordingly. The calculation increment is 0,01 m and the maximum allowable size is 2 m. The resulting flowrates and diameter are printed out for inspection. The user may investigate the effect of specified pressure drop on column diameter by specifying a new pressure drop value.

At this point the operating diagram is complete, the column diameter is calculated and the size and type of packing to be used are known. The final stage of the design is the computation of the height of packing required in each section of the column. The calculations are performed as shown in section 4.23.

195-289 'DO' loop 41 computes vapour and liquid mass transfer coefficients and interfacial surface area at each point in the stripping section of the column. The composition and flowrates at these points have been previously calculated in

OPLINE and FLOWS respectively. FLOWS has also stepped off an ideal reboiler stage. (See Chapter 3.) The check on line 195 results in a single pass through loop 41 if no stripping section exists. This sets all counters required in loop 42, and reads physical property data.

The mass transfer calculations discussed in section 4.232 are performed within loop 41 as follows. The calculations start at the feed point MPTS ($x_a y_a$ on Figure 4.1 (c)) and move down the stripping section to point MFIN ($x_1 y_1$). At each point:

197-201	Extract flow, temperature and composition data from relevant arrays. Units converted to FPS for use in Shulman's correlations.
203,204	Liquid density determined in RHOLIQ and converted to FPS units.
205	Vapour density computed.
206	Liquid viscosity determined in VISCL (units centipoise).
207-215	Choice of coefficient and exponents for Equation (4.16) based on liquid flowrate (see Appendix A, Table 2).
217,218	Check vapour density value for determining the form of Equation (4.15). See section 4.232 (c). ($\sim$2% variation allowed.)
220	a_{AW} computed using Equation (4.16).
222,223	Surface tension determined in SURTEN and converted to units of dynes/cm (required in empirical expressions for Φ_s and H*, see Appendix A, Table 1.)
224-231	Selection of empirical expression for ϕ_s, based on type of packing used. (See Appendix A, Table 1.)

232	ϕ_s computed (see Appendix A, Table 1.)
233	ϕ_{sW} " " " " " "
234	ϕ_{tW} " " " " " "
235-252	Selection of empirical expressions for H* based on type of packing and liquid viscosity. (See Appendix A, Table 1.)
253	H* computed.
254	ϕ_{0_W} computed using Equation (4.14).
255	ϕ_0 " " " (4.15).
256	ϕ_t " " " (4.13).
257	a " " " (4.12).
258	$\varepsilon_0 = \varepsilon - \phi_t$ (see section 3.232 (b)) for use in vapour phase mass transfer coefficient calculation.
259,260	Vapour diffusivity determined in DIFVAP and converted to FPS units ft^2/hr.
262,263	Vapour viscosity determined in VISCV and converted to FPS unts lbs/ft hr from centipoise.
264	Calculation of vapour phase Schmidt number.
265	Superficial molar vapour flowrate kg moles/hr m^2.
266-268	Calculation of F_V using Equation (4.11). Note that VMOLE sets F_V in SI units.
269	Vapour phase mass transfer coefficient $k_y = F_V$ (Equation (4.8)).
271-272	Liquid diffusivity determined in DIFLIQ and converted to FPS units ft^2/hr.
274,275	Liquid density from RHOLIQ, units lbs/ft .
276	Convert liquid viscosity (see line 205 above) to units of lbs/ft hr.

277 Liquid phase Schmidt number.

278-280 Calculation of k_L using Equation (4.7).

282,283 Calculation of molar density of heavy component.

284 Equation (4.10) used to determine F_L (units FPS).

285 $k_x = F_L$ (converted to SI units).

286-288 Store V,L, and a, in SI units, for use in integration procedure.

290 Return to start of 'DO' loop for each point defined on the operating line.

292-384 'DO' loop 42 performs the same calculations as loop 41, for the rectifying section starting at the top and working down $((x_2,y_2) \rightarrow (x_a,y_a))$.

Referring to Equation (4.3) and Figure 4.1 (c), values of V,L,y,x,k_y,k_x and a are known at each point in the column at the end of loop 42. The remaining unknowns are y_i and x_i.

385-389 'DO' loop 43 defines the equations of the lines of slope $-k_x/k_y$ joining points x,y and x_i,y_i in the rectifying section.

391-394 'DO' loop 44 duplicates 43 for stripping section.

396-415 'DO' loop 45 finds points of intersection of equilibrium curve and lines with slopes $-k_x/k_y$, (x_i,y_i), using the standard iterative search technique.

417-436 'DO' loop 46 duplicates 45 for the stripping section.

437 Determines which phase limits mass transfer ($k_x > k_y$ or $k_x < k_y$). The check is made at the point of maximum flowrate and the rest of the column is assumed to behave similarly. The results of this check determine the expression used in the integration.

439-451 'DO' loops 61 and 62 compute the values of the function selected by the check.

456 Subroutine SIMPS performs a numerical integration to determine the height of packing required in the rectifying section.

460-475 'DO' loop 63 and 64 duplicate 61 and 62 for the stripping section and the height of packing is computed through SIMPS.

477-478 Subroutine PACRES prints out a full set of results. Tables of mass transfer coefficients and other design data may be suppressed if they are not required.

480-500 If a pinch point is detected while computing x_i and y_i values in loop 45 and 46, a set of results is printed out that will enable the user to construct the x,y diagram to determine the position of the pinch.

501-504 A maximum diameter of 2 metres is allowed.

4.4 GENERAL COMMENTS ON PACKED

In the numerical integration procedure using Simpson's rule, the limits of integration for each section of column are as follows (refer to Figure 4.1).

Rectifying section Vy at position y_2 to Vy at ya.
Stripping section Vy at ya to Vy at y_1.

The interval between points Vy on the x axis is not necessarily constant because of the variation in flowrate caused by enthalpy considerations. It has been found, however, that for all systems encountered, the variation in flowrate is small over a short length of packing. In general, provided that at least 20 points are chosen on a particular operating line, the effect is negligible. This is consistent

with the assumption of equimolar counter diffusion in distillation systems.

4.5 SUBROUTINES ASSOCIATED WITH MAINLINE PROGRAM PACKED

The subroutines dealing with the operating diagram are discussed in Chapter 3.

4.51 Subroutine PACRES

This routine is used to print out the results of a packed column design. The format statements are self-explanatory. For further details refer to the sample printouts in Chapter 8.

Either or both of the following sets of data tables may be suppressed if the user does not require them.

(1) Composition, temperature and flowrate profiles.
(2) Mass-transfer data tables.

4.52 Subroutine PACTYP

This subroutine determines the type and size of packing to be used in the column design. If the user is providing original packing data this must be inserted in PAKING with appropriate code numbers defining its position through the GO TO statements. These code numbers may then be entered through PACTYP and the data will be processed. This is discussed further in the User Manual (see section 7.3).

The conversational output from PACTYP can be suppressed and a data element used. The format statements preceding each input explain the subroutine's text.

4.53 Subroutine PAKING

This subroutine provides the empirical data relating to the type of paking specified. The user may add data on other packings to this routine and assign them code numbers.

The first code number refers to the type of packing and the second number to the size. The required packing is selected and the data is returned through the calling list (see User Manual). The variables listed in PAKING are defined in the nomenclature.

4.54 Subroutine FDATA

This subroutine uses a least-squares technique to fit a polynomial to given set of data. The routine is called by the mainline program 'PACKED' to determine the ordinate corresponding to a specified abscissa on the flooding and pressure drop correlation chart.

Data has been taken from the curves on the chart shown in Figure 1, Appendix A. These data tables are automatically added to the runstream through the ERTRAN facility. The user specifies the allowable working pressure drop and the routine selects the corresponding correlation curve. The user may supply data for other correlation curves, if required. Full details on how this may be done are given in the User's Manual, section 7.3.

The subroutine makes use of a UNIVAC library-routine 'GJR' to solve the simultaneous equations in the least-squares procedure. Polynomials of orders one through ten are tested by the routine and the best fitting polynomial is selected for the required calculations.

Subroutine FDATA applies the least-squares fit technique as follows. The calling list includes the abscissa, ABS, and a counter specifying the correlation curve to be used, NCURVE. ORD is the ordinate value returned to the mainline on completion of the calculations.

Line No.	Program Action
6-24	Correlation curve data extracted from flooding chart is stored in data elements. The required set is selected by NCURVE.
25-28	Data read in from element.
33-50	'DO' loops 10,11 and 12 calculate the terms in the matrices AMAT and CMAT.
51-53	CMAT inserted in row N+1 of AMAT, for nth order system, as required by subroutine GJR.
55	UNIVAC subroutine GJR called to solve matrix AMAT by Gauss-Jordan reduction.
57-65	Terms of the solution matrix are extracted from row N+1 of AMAT and the sum of squared errors is computed.
66	Resultant error compared with previous value (initial value = 1000; line 30).
68	System limited to a 10th order polynomial.
71-75	Coefficients of polynomial with the best fit are stored in COEFF.
78-85	Calculation of ORD using the best fitting polynomial.
86-88	The order of the polynomial fitted, and the resultant sum of squared errors are printed out for inspection.
90-92	Error message printed out if GJR fails. JC is defined in the UNIVAC manual describing GJR.

4.55 Physical property calculations

The subroutines described in this section compute physical property data required in PACKED. The subroutines are designed for general use by programs requiring physical property data, and are not specific to PACKED. In each subroutine, program data are entered through the calling list. Any additional data required are read from a prepared data element using the ERTRAN routine. The subroutines may thus be incorporated in any program by defining the variables in the calling lists and setting up the necessary data elements.

General correlations for which data are readily available have been incorporated, and a choice of calculation procedures is offered in certain subroutines. This facilitates collation of required data tables. PACKED assumes that data returned from the various subroutines are expressed in specific units. Conversion factors are included in PACKED to adjust these units where necessary in the course of the calculations. The use of conversion factors is discussed in the User Manual.

The correlations used in each subroutine are discussed in this section and the calling list variables are defined. Details of the data elements required by the subroutines are given in the User Manual.

4.551 Subroutine RHOLIQ: This subroutine is used to compute liquid-phase pure-component and binary solution densities.

(a) Pure-component density:

Two procedures are included for determination of pure component densities at a given temperature.

(1) Extrapolation of a single value ρ_1, specified at temperature T_1, using the correlation [67]

$$\frac{(\rho_L - \rho_V)_2}{(\rho_L - \rho_V)_1} = \left[\frac{T_c - T_2}{T_c - T_1}\right]^{EXP} \qquad (4.19)$$

where the component critical temperature, T_c, must be specified, and exponent EXP depends on the type of compound considered [68]. The components are categorized by the user in the main-line program (lines 60-66 in PACKED), thus defining the appropriate exponents. Compound categories and corresponding exponents are tabulated in the User Manual.

Vapour density terms ρ_V are not included in the subroutine. This limits the subroutine to systems at total pressures below 10^3kPa.

(2) Interpolation between points in a temperature vs. density data table.

The data table is provided by the user and the subroutine interpolates using the standard linear interpolation technique described previously.

Any combination of the above two procedures is permitted. If, for example, a set of temperature-density data is available for one component and the density of the second component is known at 25°C, the subroutine will accept these data and then select procedure (2) above for the first component and procedure (1) above for the second. Data are compiled in element as shown in the User Manual (section 7.341).

(b) Solution density:

Two procedures are included for determination of solution densities at a given temperature.

(1) Combination of individual component densities as calculated above, using the following expression. For components A and B:

$$\rho_s = \frac{\rho_A \rho_B (x_A MW_A + x_B MW_B)}{(x_A MW_A \rho_B + x_B MW_B \rho_A)} \qquad (4.20)$$

(2) Interpolation between points in a composition vs. density data table with subsequent extrapolation to the correct temperature. The composition vs. density data must be quoted at a known base temperature. Critical temperatures of both components must also be supplied. The extrapolation correlation is adapted from the pure component correlation as follows [68].

For pressures below 10^3kPa

$$\rho_s = \rho_{s_b} \left[\frac{T_c' - T_2}{T_c' - T_b} \right]^{EXP'} \qquad (4.21)$$

where $T_c' = x_A T_{c_A} + x_B T_{c_B}$ (Kay's equation [58])

$EXP' = (EXP_A + EXP_B)/2$ where EXP_A AND EXP_B refer to the light and heavy components respectively.

T_b = base temperature

ρ_{s_b} = solution density obtained from composition-density data table by interpolation.

Refer to the User Manual, section 7.341 for a description of the data elements required.

The terms in the calling list are defined as follows:

(a) Supplied by calling program:

Variable	Program
NPHY	NPHY = 1 reads data through ERTRAN. NPHY > 1 bypasses read statements.
NR	NR = 1 computes solution density using method (2) above. NR ≠ 1 computes solution density using method (1) above.
NL,NH	Define exponents EXP for light and heavy components respectively in Equation (4.19) above.

TG,X	Temperature and composition (light component) of binary solution considered. (^{o}C, mole fraction)

(b) Returned to calling program

Variable	Program
RHO1,RHO2	Pure component densities of light and heavy components respectively.
RHOM	Density of binary solution.

Temperature T_c, T_1 and T_2 in the above equations are in K. Internal conversion factors assume that all supplied temperature data are in ^{o}C, so conversion by the user is unnecessary providing data are available in ^{o}C.

Density data must be supplied in g $m\ell^{-1}$ and the subroutine returns densities to the calling program in the same units. Conversion to other units must be done by the calling program where necessary. In mainline PACKED, conversion factor RHOCON converts g $m\ell^{-1}$ to lb ft^{-3}.

4.552 Subroutine SURTEN: This subroutine calculates the surface tension of binary solutions. Two calculation procedures are included.

(a) Calculation of individual component surface tensions and combination of these to yield the surface tension of the mixture by means of Stakhorsky's correlation [69].

$$\sigma_m = \sigma_A\sigma_B/[x_A\sigma_A + X_B\sigma_B] \tag{4.22}$$

Component surface tensions may be computed in one of three ways

(1) Sugden's equation

$$\sigma = \left[\frac{[P](\rho_L - \rho_V)}{MW}\right]^4 \tag{4.23}$$

where [P] is the Sugden Parachor for the component [58] and vapour density, ρ_V, is ignored. As only liquid mixture surface tensions are required, and $T < T_{bp}$, this is permissible [58]. Liquid density values, ρ_L, are provided by subroutine RHOLIQ.

(2) Extrapolation of a single temperature-surface tension data pair using Lövgren's correlation [70]

$$\sigma_2 = \sigma_1 \left[\frac{T_c - T_2}{T_c - T_1}\right]^{1,2} \qquad (4.24)$$

(3) Interpolation between values in a temperature-surface tension data table.

(b) Interpolation between values in a composition-surface tension data table with subsequent extrapolation from the base temperature to the required temperature using the correlation in (2) above. A psuedo critical temperature for the mixture is calculated using Kay's equation [58] as in RHOLIQ.

The terms in the calling list are defined in section 4.551, except for SIGM which is the binary solution surface tension returned to the calling program. NL and NH are required because RHOLIQ is called by SURTEN.

Mainline PACKED requires surface tension data in units of dynes/cm. The conversion factor SIGCON in PACKED must be appropriately set if data is provided in other units. For further details refer to the User Manual (section 7.342).

4.553 Subroutine DIFLIQ: In binary distillation systems, steady state equimolar counter-diffusion may be assumed in both the liquid and the vapour phases [9]. Liquid phase molecular diffusivities are determined by this subroutine. These values are used in PACKED to compute Schmidt numbers and mass transfer coefficients (Equation (4.7)).

Although the diffusivity of a solute in a solvent varies appreciably with concentration, the following relationship holds [58],

$$\frac{D_{AB}\mu}{T} = \text{const.} \qquad (4.25)$$

where D_{AB} refers to the diffusivity of the light component A (solute) in the heavy component B (solvent) at infinite dilution. Values of D_{AB} may be extracted from the literature or derived using an expression such as that of Wilke and Chang [71]. At infinite dilution μ is the viscosity of the pure solvent (component B) and T is the temperature at which both D_{AB} and μ are derived. Equation (4.25) does not hold for systems with high viscosities, but as such systems are seldom encountered in distillation this is not of great importance [58].

The diffusivity of a solution of A and B, D_L, at any temperature T' may be computed as

$$D_L = \frac{\text{const.}T'}{\mu_{m_L}} \qquad (4.26)$$

where μ_{m_L} is the viscosity of the solution.

Most liquid diffusivities D_{AB} fall within the range $0{,}5 \times 10^{-5}$ - $5{,}0 \times 10^{-5}$ cm^2sec^{-1}. The factor 10^{-5} is contained in subroutine and the user must therefore supply values as $D_{AB} \times 10^5$ cm^2sec^{-1}. The temperature at which D_{AB} is known, is supplied in ^oC. This is converted to K by the routine where necessary. Subroutine VISCL (section 4.555) is called to determine μ and μ_{m_L}, and the solution diffusivity is computed using Equations (4.25) and (4.26) above.

Calling list variable D_L is the liquid diffusivity, in cm^2sec^{-1}, returned to the calling program. Factor DLCON in PACKED converts the units to ft^2hr^{-1} where necessary. The other calling list variables are defined in section 4.551. Counters NL and NH are required because VISCL calls RHOLIQ. The necessary data element is described in the User Manual, section 7.343.

4.554 Subroutine DIFVAP: This subroutine computes vapour diffusivities using Arnold's correlation [72]. These values are used in PACKED to calculate vapour phase Schmidt numbers.

Assuming equimolar counter diffusion as discussed in section 4.553, Arnold's correlation is written as follows. For components A and B

$$(D_V)_{AB} = \left[\frac{0{,}00837\ b^{0,5}}{(V_A^{0,333} + V_B^{0,333})^2}\right]\left[\frac{T^{2,5}}{(T + C_{A,B})}\right] \qquad (4.27)$$

where $b = \dfrac{MW_B + MW_B}{MW_A\ MW_B}$

and Sutherland's constant $C_{A,B} = 1{,}47\ F\ (T_{bp_A} . T_{bp_B})^{0,5}$

where $F = \left[\dfrac{2(V_A^{1/3}\ V_B^{1/3})^{0,5}}{V_A^{1/3} + V_B^{1/3}}\right]^3$ [72]

P = absolute pressure of system in atmosphere.

V_A, V_B = molar volume at normal boiling point $m\ell . g\ mole^{-1}$

T_{bp_A}, T_{bp_B} = normal boiling points in K.

T = system temperature in K.

MW = molecular weight of component.

Vapour diffusivity varies only slightly with composition unless the molecular-mass ratio for the system is large [58]. In normal distillation systems this is not so. The molar volume, boiling point and molecular weight of each component must be provided by the user (see User Manual, section 7.344). Boiling point temperature is supplied in ^{o}C and converted internally. The diffusivity value is returned to the calling

program in units of cm^2sec^{-1}. Factor DVCON in PACKED converts these units to ft^2hr^{-1} where necessary.

Willee and Lee [73] showed that the average and maximum deviation obtained using this correlation are 8,4% and 20,5% respectively. The subject of vapour phase diffusivity has been more extensively investigated than liquid phase diffusivity and the correlations may be applied with more confidence [58]. Although the procedure described in section 4.553 achieves reasonable results, it can not be considered as rigorous as the above correlation. Nevertheless, both procedures are sufficiently accurate for the requirements of the program PACKED.

The calling list variables not defined in section 4.551 are PT, the total pressure of the system in mm Hg, and DV, the vapour diffusivity value returned to the calling program.

4.555 Subroutine VISCL: This subroutine calculates the solution, and individual component, liquid viscosities for binary systems. Solution viscosities are computed using the component viscosity values and the following correlation [74],

$$\mu_{m_L} = [x_1\mu_1^{1/3} + x_2\mu_2^{1/3}]^3 \qquad (4.28)$$

Component viscosities are calculated in one of two ways:

(1) Thomas's equation [75]:

$$\mu = 0{,}1167\ \rho_L^{\frac{1}{2}}\ 10^{\gamma} \qquad (4.29)$$

where $\gamma = B(\frac{1-T_r}{T_r})$

and B is a viscosity constant.

The viscosity constant and critical temperature for each component must be specified to use this option (see User Manual, section 7.345). Equation (4.29) holds for $T_r \leq 0{,}7$ [75].

(2) Interpolation between values in a temperature-viscosity table.

The calling list is similar to that of RHOLIQ. AMUA, AMUB and AMUM are the pure component and binary solution liquid viscosities returned to the calling program. Data are returned to PACKED in units of centipoise. Factor AMLCON converts the units to lb/ft hr. where necessary. For further details refer to the User Manual, section 7.345.

4.556 Subroutine VISCV: This subroutine calculates pure component, and binary mixture, vapour-phase viscosities. The choice of calculation procedures is the same as in subroutine VISCL, with different correlations used.

The vapour mixture viscosity is computed using the following equation [76],.

$$\mu_{m_V} = \frac{\Sigma y_i \mu_i \ MW_i^{0,5}}{\Sigma \ y_i \ MW_i^{0,5}} \qquad (4.30)$$

The component viscosities are calculated either by interpolation of temperature-viscosity data or by means of Arnold's correlation [77].

For component i

$$\mu_i = \frac{27,0.MW_i^{0,5}T^{1,5}}{V_i^{2/3} \ (T + 1,47 \ T_{bp_i})} \qquad (4.31)$$

where V_i is the molar volume in mℓ g mole^{-1}.

The calling list variables not defined in section 4.551 are Y, the vapour composition (light component) and AMUM, the binary mixture vapour viscosity. The viscosity value is computed in micropoise in the above correlations and these units are converted to lb/ft hr in PACKED by factor AMVCON, where required. (See User Manual, section 7.346.)

4.557 Subroutine RHOH2O: Density of water is well documented [58] and this subroutine requires a table of NDATA temperature-density data pairs. The data element is H2ORHO and the required value is found by linear interpolation. The larger the data table, over a set temperature range, the more accurate the value obtained.

The calling list variables are:

NPHY: defined in section 4.551.
T : system temperature in ^{o}C.
RHO : density of water at temperature T returned to the calling program in g $m\ell^{-1}$.

4.6 SUMMARY OF BINARY PACKED COLUMN DESIGN PROGRAM

Program name: PACKED

Type: Mainline with subroutines.

Operation mode: Demand, on UNIVAC 1106 with EXEC 8.

1. Title: Rigorous design of packed columns for binary continuous distillation.

2. Solution type: Design program.

3. Application: PACKED can be used to design packed columns for binary continuous distillation application. A Ponchon-Savarit operating diagram is used and the design procedure of Treybal, using Shulman's mass transfer analysis, is adopted. The program will compute the column diameter for a given pressure drop and feed rate or will indicate the coordinates of a point on the flooding and loading correlation chart of the U.S. Stoneware Co. [9], if a diameter is specified. The height of packing required in the rectifying and stripping sections of the column is computed using a Simpson's rule numerical integration procedure.

4. Program Development:

(a) Computer Used: UNIVAC 1106 with EXEC 8 control.

(b) Minimum Core Requirements: 23 K

(c) Language: FORTRAN IV with modifications allowed on the above system.

(d) Typical run time: 8 secs

CHAPTER 5

MULTICOMPONENT DISTILLATION

5.1 INTRODUCTION

Solving multicomponent distillation problems manually is tedious and time-consuming because of the large number of repetitive calculations required. As a result students are usually unable to do more than one or two worked examples in the time allotted to this section of the undergraduate course and emphasis is placed on calculation technique rather than the significance of the results obtained. The iterative calculations necessary for the rigorous design of multicomponent systems are best performed on a computer, thus reducing the time required to solve a problem from hours to seconds. With a computer program available for solving problems, students would have more time for analysing the results and would thus be able to investigate multicomponent distillation in more detail. This is discussed further in Chapter 8.

The Thiele-Geddes calculation procedure used in this study simulates the performance of known distillation columns. Simulation can be used for design by successive applications, with adjustment of variables until product specifications are met. A single solution will determine the top and bottom compositions achieved in a given column configuration. The program developed here will enable students to determine the effects of a number of parameters on a multicomponent system.

Only thermodynamically ideal systems are considered, as in the binary processes discussed previously. Equilibrium data are considered in terms of equilibrium constants rather than relative volatilities, and the program includes facilities for investigating enthalpy and tray efficiency effects. The

calculation procedures used in the program correspond to material presented in the undergraduate course.

5.2 THEORETICAL BACKGROUND TO THE THIELE-GEDDES METHOD

5.21 Multicomponent systems

The basic relationships describing an equilibrium stage (see Chapter 1) can be written for a multicomponent distillation column as follows: Referring to Figure 5.1, for each component i on tray j:

$$\text{Equilibrium constant } K_{i,j} = y_{i,j}/x_{i,j} \tag{5.1}$$

Overall balances:

$$\text{Material} \quad FZ_{i,F} = Dx_{i,D} + Bx_{i,B} \tag{5.2}$$

$$\text{Enthalpy} \quad FH_F = Q_C - Q_R + DH_D + Bh_B \tag{5.3}$$

where $$H_F = \frac{V_F}{F} \sum_{i=1}^{NC} H_{i,F}y_{i,F} + \frac{L_F}{F} \sum_{i=1}^{NC} h_{i,F}x_{i,F} \tag{5.4}$$

Rectifying section:

$$\text{Material} \quad V_n y_{i,n} = L_{n+1}x_{i,n+1} + Dx_{i,D} \tag{5.5}$$

$$\text{Enthalpy} \quad V_n H_n = L_{n+1}h_{n+1} + DH_D + Q_C \tag{5.6}$$

where $$H_n = \Sigma y_{i,n}H_{i,n} \tag{5.7}$$

$$h_n = \Sigma x_{i,n}h_{i,n} \tag{5.8}$$

Stripping section:

$$\text{Material} \quad L_{m+1}x_{i,m+1} = V_m y_m + Bx_{i,B} \tag{5.9}$$

$$\text{Enthalpy} \quad L_{m+1}h_{m+1} = V_m H_m + Bh_B - Q_R \tag{5.10}$$

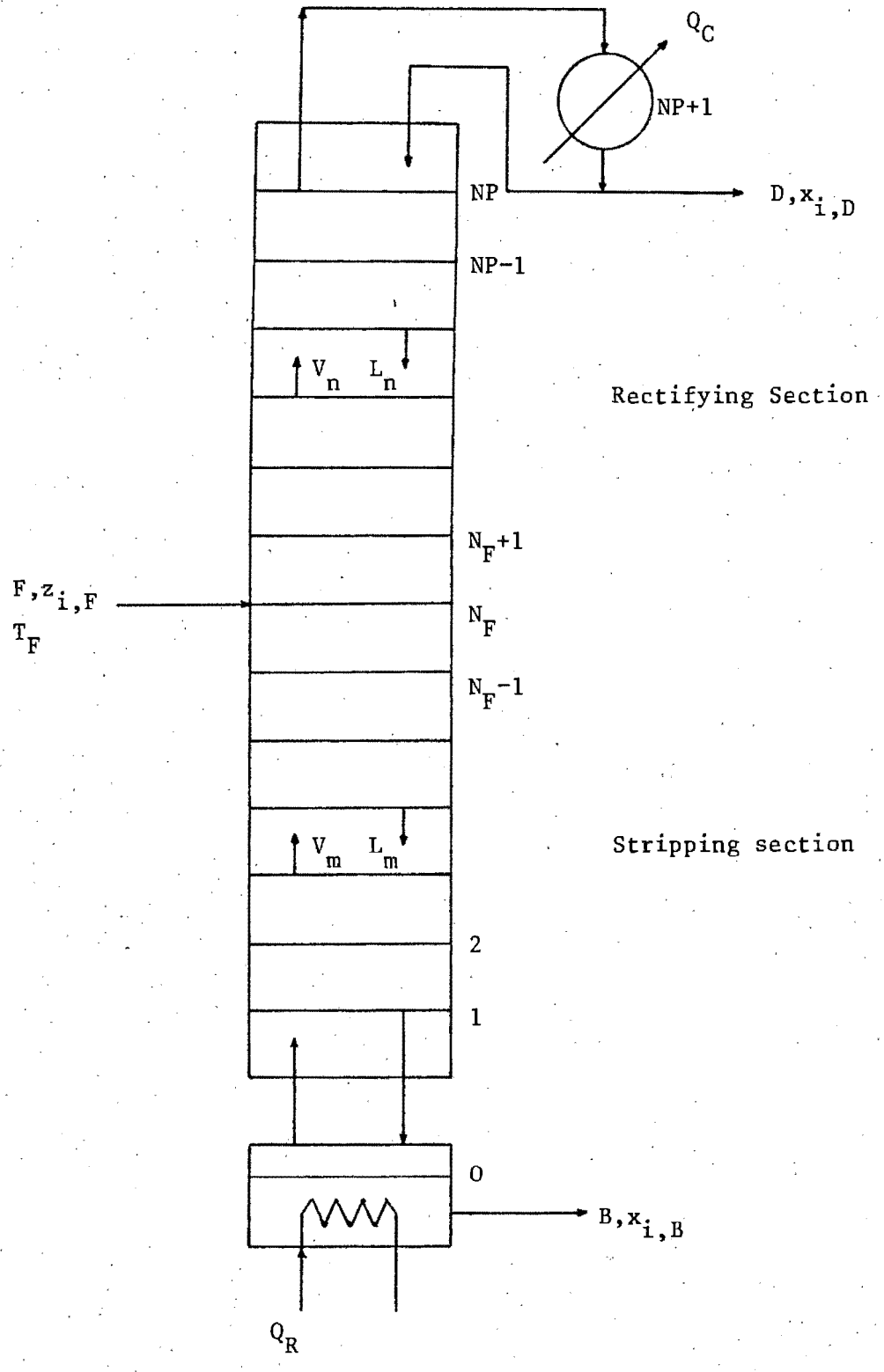


FIGURE 5.1 MULTICOMPONENT DISTILLATION COLUMN

Writing these equations for two components only, reduces them to the binary system equations shown in section 3.2.

5.22 General Specifications

The following parameters must be specified or calculated before the iterative calculations begin.

(1) Number of theoretical trays in the column.
(2) Position of feed tray.
(3) Composition, condition and flowrate of feed.
(4) Distillate withdrawal rate.
(5) Reflux ratio.

The feed condition is computed from temperature or enthalpy specifications by means of a flash calculation routine. For problems in which constant molal overflow is assumed, a q-value must be specified if the feed is super-cooled or super-heated. (See section 5.54.)

5.23 Initial Assumption and Estimates

The Thiele-Geddes method adopts temperature on each tray as the independent variables. Thus an initial estimate of the temperature profile is required before calculations are begun. A linear profile, based on the condenser and reboiler temperatures, has been found to be a suitable estimate. If condenser and reboiler temperatures are not available the boiling points of the light and heavy key components, or some reasonable estimate of top and bottom temperatures, may be used.

The initial vapour and liquid profile through the column must also be determined. Many tutorial problems assume constant molal overflow, and in such instances the intial flowrate profile will be unaffected by subsequent calculation. For problems that include enthalpy consideration it is customany to use the assumptions of constant molal overflow to

find an initial flowrate profile. It is also advantageous to ignore enthalpy effects for the first two iterations as this allows the system to stabilize.

5.24 Description of the Thiele-Geddes Algorithm

In describing the calculation procedure it is convenient to define several symbols that simplify the material balance and equilibrium expressions. The objective is to develop working equations that are not explicit functions of composition because, in a simulation procedure, stream compositions are in general, unknowns.

In the rectifying section the molal flowrates of component i, in the vapour and liquid leaving tray n and in the distillate product, are defined respectively by the three expressions (see Figure 5.1):

$$v_{i,n} = V_n y_{i,n} \qquad n=f,f+1....N \qquad (5.11)$$

$$l_{i,n} = L_n x_{i,n} \qquad n=f+1,f+2,...N \qquad (5.12)$$

$$d_i = D x_{i,D} \qquad (5.13)$$

Similarly, for the stripping section, where the bottom product replaces the distillate

$$v_{im} = V_m y_{i,m} \qquad (m= 0,1,2,....f-1) \qquad (5.14)$$

$$l_{i,m} = L_m x_{i,m} \qquad (m=1,2,.....f) \qquad (5.15)$$

$$b_i = B x_{i,B} \qquad (5.16)$$

From the above definitions, the liquid leaving the feed tray, f, is included in the stripping section while the vapour

leaving f is included in the rectifying section. It is assumed that the streams leaving the feed tray are in equilibrium, as are the streams leaving all other trays.

The equilibrium relationship (Equation (5.1)) for tray n in the rectifying section may be rearranged, multiplied through by L_n and V_n/V_n, and written in terms of molal flowrates as

$$l_{i,n} = \frac{L_n}{V_n K_{i,n}} \cdot v_{i,n} \tag{5.17}$$

The term $L_n/V_n K_{i,n}$ is defined as the"absorption factor",

$$A_{i,n} = \frac{L_n}{V_n K_{i,n}} \tag{5.18}$$

and is the molar ratio of component i leaving tray n in the liquid and vapour phase. For a partial condenser an equilibrium stage is assumed and

$$A_{i,N+1} = \frac{L_{N+1}}{D\ K_{i,N+1}} \tag{5.19}$$

For a total condenser

$$A_{i,N+1} = \frac{L_{N+1}}{D} \tag{5.20}$$

Similarly, for tray m in the stripping section, the equilibrium expression may be written in terms of molal flowrates as

$$v_{i,m} = \frac{V_m\ K_{i,m}}{L_m}\ l_{i,m} \tag{5.21}$$

where $V_m\ K_{i,m}/L_m$ is the "stripping factor".

$$S_{i,m} = \frac{V_m\ K_{i,m}}{L_m} \tag{5.22}$$

For the reboiler, an equilibrium stage

$$S_{i,o} = \frac{V_o K_{i,o}}{B} \tag{5.23}$$

These definitions of molal flowrates, absorption factors and stripping factors can be used to modify the equilibrium expression for each tray (Equation (5.1)) and general material balance expressions for each section (Equation (5.5) and (5.9)) as follows:

Rectifying section

Condenser Equilibrium $$\frac{l_{i,N+1}}{d_i} = A_{i,N+1} \tag{5.24}$$

Tray Equilibrium $$\frac{l_{i,n}}{d_i} = A_{i,n} \frac{v_{i,n}}{d_i} \quad n=f+1,f+2\ldots N \tag{5.25}$$

Material balance $$\frac{v_{i,n}}{d_i} = \frac{l_{i,n+1}}{d_i} + 1 \quad n=f,f+1,\ldots N \tag{5.26}$$

Stripping section

Reboiler Equilibrium $$\frac{v_{i,o}}{b_i} = S_{i,o} \tag{5.27}$$

Tray Equilibrium $$\frac{v_{i,m}}{b_i} = S_{i,m} \frac{l_{i,m}}{b_i} \quad m=1,2,\ldots,f-1 \tag{5.28}$$

Material balance $$\frac{l_{i,m+1}}{b_i} = \frac{v_{i,m}}{b_i} + 1 \quad m=0,1,2,..f-1 \tag{5.29}$$

The unknowns d_i and b_i in Equations (5.5) and (5.9) have been eliminated by dividing through appropriately to yield the above expressions. The absorption and stripping factors are computed for each component on each tray using the initial temperature profile to compute equilibrium constants, and the initial vapour and liquid flowrate data. The tray-to-tray calculations then proceed as follows:

(i) Rectifying section

Calculations are initiated at the condenser. The type of condenser specified determines the form of $A_{i,N+1}$ and Equation (5.24) computes the starting value $l_{i,N+1}/d_i$. This is substituted in Equation (5.26) to yield $v_{i,n}/d_i$. Equations (5.25) and (5.26) are then used alternately in working down the column to the feed tray until values $l_{i,f+1}/d_i$ and $v_{i,f}/d_i$ are determined.

(ii) Stripping section

The successive substitution procedure is the same as for the rectifying section. Calculations are initiated at the reboiler (Equation (5.27) and Equations (5.28) and (5.29) are used in working up the column to determine $v_{i,f-1}/b_i$ and $l_{i,f}/b_i$.

At this point the first iteration is complete and the degree of matching achieved at the feed tray is determined.

5.241 Feed tray analysis: The above assumption of feed tray equilibrium conditions implies that the feed stream entering at the feed tray position achieves instantaneous equilibrium with the solution on the feed tray, resulting in equilibrium streams leaving the feed tray. This is an extremely idealized situation, but is one that has found wide acceptance as an approximation of a real system [12,21,31]. (If, in practice, the vapour fraction of the feed is sparged just below the actual feed tray so that it mixed well with the vapour rising from the stripping section, and the liquid portion of the feed is added to the liquid in the downcomer from tray f+1, the possibility of achieving equilibrium streams leaving the feed tray would be improved.)

The equilibrium expression for the feed tray may be written in terms of the previously defined parameters as

$$l_{i,f} = A_{i,f}\, v_{i,f} \tag{5.30}$$

where $A_{i,f} = L_f/V_f K_{i,f}$ (5.31)

and $V_f = V_{f-1} + (1-q)F = V_{f-1} + V_F$ (5.32)

$$L_f = L_{f+1} + qF = L_{f+1} + L_F \tag{5.33}$$

The feed condition, q, is determined by flash calculation (see section 2.263). In a constant molal overflow system, V_f is the rectifying section vapour rate and L_f is the liquid rate in the stripping section.

Multiplying and dividing Equation (5.30) by d_i and b_i, and rearranging, yields,

$$\frac{b_i}{d_i} = A_{i,f}\,\frac{b_{i,f}/d_i}{l_{i,f}/b_i} \tag{5.34}$$

The results of the tray-to-tray calculations are used to determine the ratios b_i/d_i. Values of b_i and d_i are then computed. The overall material balance, Equation (5.2), is rearranged and written in terms of molal flowrates to define d_i as

$$d_i = \frac{FZ_{i,F}}{1 + b_i/d_i} \tag{5.35}$$

Values of b_i are then computed using the expression

$$b_i = d_i\left(\frac{b_i}{d_i}\right) \tag{5.36}$$

The more obvious rearrangement of Equation (5.35), $b_i = FZ_i d_i$ is to be avoided because of the inaccuracies involved in computed differences when d_i and $FZ_{i,F}$ are almost equal, as

in the case of light components [12].

The computed values of b_i and d_i are then used to calculate the values of $v_{i,n}$, $l_{i,n}$, $v_{i,m}$ and $l_{i,m}$, through the column, from the ratios defined by Equations (5.24)-(5.29). In general the sum of the component molal flowrates thus calculated will not equal the originally calculated total flowrate leaving a particular tray.

Similarly the summed values of b_i or d_i will not equal B or D respectively, unless the temperature profile originally chosen is the correct one. A new temperature profile is obtained by calculating the compositions of the streams leaving each tray and computing the bubble point of the liquid or the dew point of the vapour for these composition values.

Composition on any tray j can be computed using the following expression

$$x_{i,j} = \frac{l_{i,j}}{\Sigma l_{i,j}} \tag{5.37}$$

$$y_{i,j} = \frac{v_{i,j}}{\Sigma v_{i,j}} \tag{5.38}$$

Product compositions:

$$x_{i,D} = \frac{d_i}{\Sigma d_i} \tag{5.39}$$

$$x_{i,B} = \frac{b_i}{\Sigma b_i} \tag{5.40}$$

When convergence is obtained the following relationship will hold:

$$\Sigma d_i = D \tag{5.41}$$

$$\Sigma b_i = B \tag{5.42}$$

$$\Sigma l_{i,j} = L_j \tag{5.43}$$

$$\Sigma v_{i,j} = V_j \tag{5.44}$$

Thiele and Geddes did not propose a technique for improving the rate of convergence, and the method of direct iteration used by them can be extremely time-consuming. A convergence procedure is therefore required, and the Holland θ method is used in this study.

5.25 Holland's θ method for convergence

5.251 The θ method: The procedure proposed by Lyster et al. [33] considerably increases the rate of convergence of the Thiele-Geddes algorithm. The primary function of a convergence technique is to improve the estimate of the independent variables used in each iteration, thus reducing the total number of iterations required to reach a solution. The θ method is based on the fact that when the temperature profile is not correct, the b_i/d_i ratios (Equation (5.34)) will all generally be either too high or too low, with the result that, through Equation (5.35), Equation (5.41) will not hold.

Following Lyster's procedure, the calculated b_i/d_i ratios are corrected by some factor θ such that the sum of the d_i's, calculated using Equation (5.35), equals the specified distillate withdrawal rate (Equation (5.41)). These adjusted d_i values are then used, as described above, to calculate the molal flowrates, the liquid and vapour compositions, and finally, the new temperature profile.

This technique, combined with the Thiele-Geddes calculation procedure, is recognised as one of the fastest methods for solving multicomponent distillation problems [34,35].

Mathematically the θ method can be described as follows:

The "calculated" b_i/d_i ratios are determined using Equation (5.34) and these ratios are "corrected" by the expression

$$\left[\frac{b_i}{d_i}\right]_{co} = \left[\frac{b_i}{d_i}\right]_{ca} \cdot \theta \qquad (5.45)$$

Equation (5.35) is rewritten in terms of the corrected ratios as

$$(d_i)_{co} = \frac{FZ_{i,F}}{1 + \theta (b_i/d_i)_{ca}} \qquad (5.46)$$

The method of computing θ is discussed below.

The corrected value of b_i is calculated using the expression

$$(b_i)_{co} = (d_i)_{co} \left[\frac{b_i}{d_i}\right]_{co} \qquad (5.47)$$

These corrected values are used, as previously described, to compute molal flowrates through the column.

5.252 Calculation of θ: θ is to be evaluated such that

$$\Sigma (d_i)_{co} = D_{spec} \qquad (5.48)$$

i.e. $$\Sigma \left[\frac{FZ_{i,F}}{1 + \theta (b_i/d_i)_{ca}}\right] - D_{spec} = 0 \qquad (5.49)$$

Equation (5.49) is solved for θ using Newton's method, previously described in section 2.52.

Written in terms of θ, Newton's iterative equation is:

$$\theta_{k+1} = \theta_k - \frac{f(\theta_k)}{f'(\theta_k)} \tag{5.50}$$

where

$$f(\theta_k) = \Sigma \left[\frac{FZ_{i,F}}{1 + \theta_k (b_i/d_i)_{ca}} \right] - D \tag{5.51}$$

$$f'(\theta_k) = - \Sigma \left[\frac{(b_i/d_i)_{ca} \, FZ_{i,F}}{(1 + \theta_k (b_i/d_i)_{ca})^2} \right] \tag{5.52}$$

Convergence is achieved when $f(\theta_k)$ approaches zero within a specified limit of accuracy.

Overall convergence of the solution procedure, indicating the choice of the correct temperature profile, is obtained when θ approaches unity within a specified limit of accuracy.

The application of the convergence procedure is discussed in detail in subroutine THECAL (section 5.58). The convergence characteristics of the θ method are discussed below.

5.253 Convergence Characteristics of the θ method: It will be shown (1) that the θ method adjusts the temperature profile in the correct direction for achieving convergence, and (2) that the correction achieved per iteration is greater than that achieved by direct iteration procedures.

(1) Consider a system in which constant molal overflow may be assumed. From Equation (5.26) the ratio $v_{i,n}/d_i$ is proportional to the absorption factor and thus inversely proportional to the equilibrium constant $K_{i,n}$ and hence temperature. Similarly from Equation (5.29) $l_{i,m}/d_i$ is proportional to temperature. From Equation (5.34) therefore the b_i/d_i ratios will decrease with an increase in temperature.

Thus if the temperature profile is too high the b_i/d_i ratios will be too low and the corrected ratios must be larger than the calculated ratios. Hence, by Equation (5.45), θ must be greater than unity in this instance.

On inspection of Equation (5.49) this equality will be satisfied for b_i/d_i ratios that are too low, if θ is positive and greater than unity. Similarly, if the temperature profile is too low, the b_i/d_i ratios will be too high and θ must be positive and less than unity in order to satisfy Equation (5.49).

Inspection of Equation (5.49) also shows that negative values of θ can lead to $f(\theta) \to \infty$ and that the overall material balance is destroyed. A qualitative plot of $f(\theta)$ vs positive values of θ, Figure 5.2, shows that $f(\theta) = B$ at $\theta = 0$ [12].

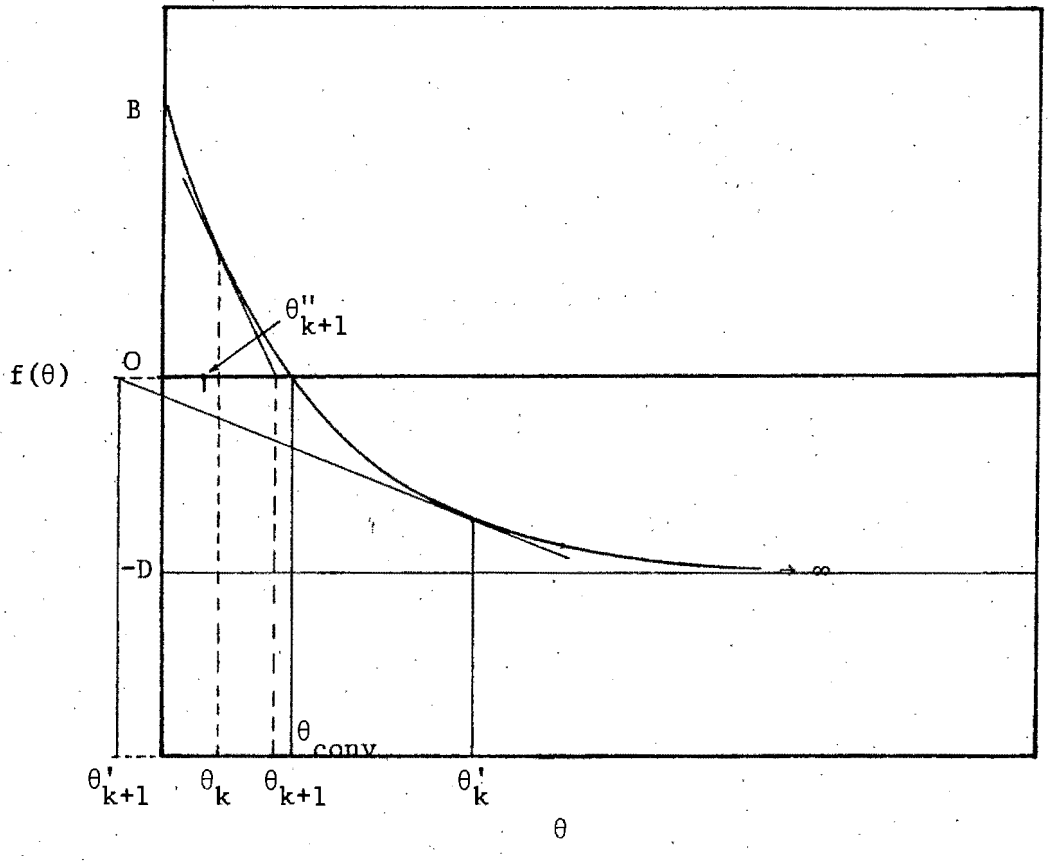


FIGURE 5.2 NEWTON ITERATION FOR θ

(from Brian [12], p.245)

The curve of $f(\theta)$ vs θ is a monotonically decreasing function for values of θ greater than 0. In applying Newton's method of tangents, if the procedure is started at $\theta = 0$ convergence will be achieved with no negative values of θ occurring. θ_{k+1} and θ_k for the (k+1)th iteration are shown in Figure 5.2. The number of iterations required for convergence can be reduced, however, by starting with a value of θ closer to 1,0. To this end the previous converged value of θ is adopted as the starting value for all overall iterations other than the first. For most problems, providing that a reasonable initial temperature profile is chosen, the value of θ seldom exceeds 2,0 and the above procedure is satisfactory. In certain situations, however, θ occasionally becomes extremely large. In such cases it was found that the rate of convergence could be improved by restructing to a maximum value of 10 000. To guard against the possibility of negative values of θ resulting during the application of Newton's method, a check is made on each iteration.

The manner in which a negative value could occur is shown in Figure 5.2, by θ'_k and θ'_{k+1}. Assume that the temperature profile and other factors caused the Newton procedure to converge at $\theta = \theta'_k$. In this case the $f(\theta)$ vs θ curve would be such as to intersect $f(\theta) = 0$ at $\theta = \theta'_k$. The next overall iteration would use θ'_k to adjust the whole system, which would result in a new $f(\theta)$ vs θ curve, such as that shown in Figure 5.2. If $\theta'_k \leq 9999$ the θ calculation routine will adopt θ'_k as the starting value of θ. The first iteration will result in θ'_{k+1} which as shown in Figure 5.2 is negative. In this event, the negative value is multiplied by -1 (see section 5.58) and the calculations are continued. This effectively restarts the procedure from point $\theta''_{k+1} = -1.\theta'_{k+1}$ shown in Figure 5.2.

By inspection, it is unlikely that a single overall iteration will cause the system to adjust itself sufficiently for negative values of θ to occur. In fact no examples of negative values were encountered in any of the problems tested, although several systems caused values of θ greater than 10 000. Thus, by restricting θ to positive finite values, the temperature profile is adjusted in the correct direction for convergence.

(2) A convergence procedure must also, by definition, improve the rate of convergence.

Consider a system containing very light and very heavy components. For a very light component, b_i/d_i will be very small and for a very heavy component b_i/d_i will be very large. Thus Equation (5.46) reduces to the following forms for each case respectively.

$$\text{Very light component } (d_i)_{co} \simeq FZ_{i,F} \tag{5.53}$$

$$\text{Very heavy component } (d_i)_{co} \simeq \frac{FZ_{i,F}}{\theta(b_i/d_i)_{ca}} \tag{5.54}$$

Equations (5.37) and (5.38) can be written as follows when the θ method of convergence is applied.

$$x_{i,n} = \frac{\dfrac{l_{i,n}}{(d_i)_{ca}} \cdot (d_i)_{co}}{\Sigma\left[\dfrac{l_{i,n}}{(d_i)_{ca}} \cdot (d_i)_{co}\right]} \qquad i = 1,NC \tag{5.55}$$

$$y_{i,n} = \frac{\dfrac{v_{i,n}}{(d_i)_{ca}} \cdot (d_i)_{co}}{\Sigma\left[\dfrac{v_{i,n}}{(d_i)_{ca}} \cdot (d_i)_{co}\right]} \qquad i = 1,NC \tag{5.56}$$

Considering Equation (5.54) above, θ will result in a greater or smaller concentration of heavy component for values less than or greater than unity respectively. Consider the case where the temperature profile for the previous trial was too low. From part (1) above, for low temperature profiles $\theta < 1,0$ and therefore the proportion of heavy component will be relatively higher (Equation (5.54). Corrected compositions calculated from Equations (5.55) and (5.56) show a greater proportion of heavy component than uncorrected compositions from (5.37) and (5.38). Hence the temperature profile computed from these compositions will be higher than that computed from compositions calculated using Equations (5.37) and (5.38), which apply to the direct iteration procedure.

The control of overcorrection of the temperature and flowrate profiles is discussed in section 5.56.

5.26 Supplementary Computation

5.261 Multicomponent vapour-liquid equilibrium: For ideal systems the vapour-liquid equilibrium relationships are expressed in terms of Henry's law as shown in Equation (5.1). Two calculation methods are available within the program for determining the equilibrium constants, $K_{i,j}$.

(i) Antoine's equation:

When ideal-solution behaviour is closely approached and the system pressure is less than 10^3 kPa, the K-values may be determined, using Raoult's Law, as the ratio of the component vapour pressure and the system total pressure (Equation (2.7)). The vapour pressure of each component is determined from Antoine's equation (Equation (2.9)).

(ii) Polynomials in temperature:

If K-value data are available in terms of temperature, at the system pressure, a polynomial can be fitted to these data and the coefficients supplied to the program. A number of authors have adopted this procedure and three different

polynomials have been included in the program. The polynomials were extracted from the work of Holland [21] and Seppala et al. [18] and are shown in the User Manual (section 7.433). These polynomials were chosen because corresponding coefficients for a number of systems suitable for tutorial problems are quoted in the literature (see Chapter 8).

The K-value calculation procedure selected by the user determines the form of the bubble point and dew point calculation. Newton's method is used to achieve convergence to the correct temperature in each case. The functions used in the calculations are discussed in section 5.42 and 5.44.

5.262 Enthalpy calculations: In the undergraduate course on distillation, the effect of enthalpy on multicomponent systems is often considered in theory only. Owing to the lengthy nature of most multicomponent problems, enthalpy balances are seldom included in manual solutions. Thus a student's practical experience of enthalpy effects is usually limited to the use of the Ponchon-Savarit procedure for solving binary problems. Enthalpy balances are included as an optional facility in the multicomponent program described here. The user is thus able to compare results obtained with and without these calculations.

The enthalpy balances are expressed by Equations (5.3), (5.6) and (5.10). These expressions are used to determine condenser and reboiler heat duties and stream enthalpies through the column. Enthalpy data are also required for an isenthalpic flash calculation (see section 5.263). The user may either provide data for the standard enthalpy subroutine or include a separate subroutine designed for a specific application.

(i) Standard enthalpy subroutine, ENERGY:

The standard enthalpy routine included in the package shares a COMMON block with the mainline line program and uses data read by subroutine READIN at the start of the run. (See section 5.41.) Vapour and liquid component enthalpies are determined from polynomials in temperature for which coefficients must be supplied. The polynomials were taken from the work of Holland [21] and Seppala [18] for reasons discussed in section 5.261. (See also Chapter 8.)

(ii) User subroutine, ENTHAL:

If the user provides an enthalpy routine it must conform to certain specifications set out in the User Manual. The calculation sequence is the same as that in the standard routine and any data required is read through the ERTRAN facility. Communication with the mainline program is through a calling list. This facility enables the user to adopt any convenient means of representing enthalpy data in the solution procedure.

A sample routine is included in the package for illustrative purposes. This routine uses enthalpy functions proposed by Kobel [55], as discussed in Chapter 3, for which coefficients for a number of compounds are available [57].

For the purposes of undergraduate tutorial problems the two enthalpy subroutines provided in the package will probably suffice. A user may wish to include a more rigorous procedure when considering real systems involving non ideal solutions, or other specific problems. It is for applications such as these that the User-routine option is included.

5.263 Flash calculation: In general, the proportions of liquid and vapour in a feed stream will not be known. At best, information on the flowrate, composition, temperature and pressure of the feed will be available. The proportions and compositions of each phase that will exist as the feed

enters the column, at the pressure of the feed tray, are determined using a flash calculation routine QFLASH. The specified feed temperature is taken as the flash temperature and the feed tray pressure is assumed to be the flash pressure. two cases are considered:

(1) Pressure of feed = feed tray pressure

(2) Pressure of feed > feed tray pressure.

For case (1) no physical flash operation is necessary, but the calculation is performed in order to determine the proportions of liquid and vapour in the feed and the corresponding composition of each phase.

Case (2) would in general require equilibrium and enthalpy data to be provided at the feed pressure. These data would be used to determine the proportion of vapour and liquid in the feed at the higher pressure, and hence the enthalpy of the feed. An isenthalpic flash calculation would then determine the vapour-liquid ratio at the flash temperature and pressure. As the details of sophisticated flash calculations are not relevant at the level of the undergraduate distillation course, it is assumed that the feed enthalpy is known at the elevated pressure.

An isenthalpic flash calculation then determines the q-value and temperature of the feed, at the feed tray pressure. Case (1) Having established that the feed temperature lies between the bubble and dew points the following material balance can be written:

$$FZ_{i,F} = V_F y_{i,F} + L_F x_{i,F} \qquad (5.57)$$

Substituting the equilibrium relationship and rearranging,

$$x_{i,F} = \frac{FZ_{i,F}}{V_F K_{i,F} + L_F} \qquad (5.58)$$

where $K_{i,F}$ is computed at the feed stream temperature. Substituting for V_F and L_F (Equations (5.32) and (5.33)),

$$x_{i,F} = \frac{z_{i,F}}{K_{i,F} - qK_{i,F} + q} \tag{5.59}$$

The flash calculation now involves finding a value of q such that: $\Sigma x_{i,F} = 1,0$. The calculation is performed by the standard iterative search technique described in section 2.55.

If the initial bubble point-dew point calculations show that the feed is in fact either superheated or supercooled at the temperature and pressure of the flash, the q-value is determined using the following expression

$$q = \frac{H_f - H_F}{H_f - h_f} \tag{5.60}$$

where H_f = enthalpy of a saturated vapour feed;
h_f = " " " " liquid "
H_F = " of feed stream at flash conditions.

Case (2): An isenthalpic flash calculation involves finding the temperature of the feed at the flash pressure and hence finding q by flash calculation. If a two phase equilibrium is found to exist at the flash conditions the procedure is as follows.

An estimate of the equilibrium vapour-liquid ratio is made, say 0,5, as an initial estimate. Using Equation (5.58) above and an iterative procedure, the temperature for this ratio is calculated at the flash pressure. Hence the phase compositions may be computed and the enthalpy at these conditions calculated. This enthalpy value is compared with the specified feed enthalpy and the estimate of the vapour-

liquid ratio is adjusted suitably. The process is then repeated until a match of enthalpies is obtained.

In the case of a superheated or a supercooled feed at flash conditions, the temperature is computed by direct iteration on the specified feed enthalpy.

5.264 Bubble Point, Dew Point and Tray Efficiency calculation: Subroutines BUBCAL and DEWCAL compute multicomponent bubble points and dew points respectively, using Newton's method of convergence as described in section 2.52. Four calculation procedures are included in each subroutine corresponding to the four equilibrium data correlations described in section 5.261. The computational techniques used in each case are described in section 5.55.

In this study multicomponent tray efficiency is considered in terms of component vaporization efficiency factors as applied by Holland [21]. This has been discussed in Chapter 1.

A component vaporization efficiency factor is expressed by Holland as

$$E_{V_{i,j}} = \frac{y^*_{i,j}}{y_{i,j}} \qquad (5.61)$$

where $y_{i,j} = K_{i,j} \cdot x_{i,j}$ is the ideal equilibrium composition, and $y^*_{i,j}$ is the composition of the vapour leaving tray j.

In the absence of any realistic procedure for calculating these efficiency factors mathematically, they must be determined from experimental data [22]. The efficiency factors are included in the tray-to-tray calculation procedure by means of modified absorption and stripping factors, defined as [21]

$$A_{i,n} = \frac{L_n}{E_{V_{i,n}} K_{i,n} V_n} \tag{5.62}$$

$$S_{i,m} = \frac{E_{V_{i,m}} K_{i,m} V_m}{L_m} \tag{5.63}$$

Computationally, this reduces to modifying the equilibrium constants defined by Equation (5.1). This is done in subroutine EQUCON where the equilibrium constants are calculated using the ideal-system data supplied by the user, and then multiplied by the specified efficiency factors. If efficiency data are not supplied these factors are set equal to 1,0 (see section 5.51).

In effect, the use of efficiency factors in this way imposes a state of psuedo-equilibrium of the system and the bubble point and dew point calculations must account for this. If the temperature profile is computed using ideal equilibrium data (Equation (5.1)) when the vapour-liquid equilibrium is expressed as

$$y^*_{i,j} = E_{V_{i,j}} K_{i,j} x_{i,j} \tag{5.64}$$

for $E_{V_{i,j}} \neq 1,0$, the tray temperatures will, in general, be either too high or too low. This results in θ converging to a value other than 1,0. This inconsistency is avoided by incorporating efficiency factors in subroutines BUBCAL and DEWCAL. (See section 5.55.)

5.3 DESCRIPTION OF MAINLINE PROGRAM MULCOM

5.31 General structure of MULCOM

The principal function of the mainline routine, in the multicomponent distillation program, is to link the various calculation steps together and to determine which options

the user intends to employ. Each stage of the Thiele-Geddes algorithm is contained in a separate subroutine. The input of initial data and the output of results are also performed through subroutines. This has been done to simplify the flowcharting and thus facilitate any future alterations.

5.32 Operation of the program

The program has been developed for operation in demand mode from a computer terminal. It can also be operated in batch mode, but this requires considerable care in setting up the runstream. In demand mode data may be supplied either conversationally or by means of a data element, as described for the binary programs.

A skeleton flowchart of MULCOM is shown in Figure 5.3. The user may solve a problem several times within a single execution for different values of certain variables (see User Manual). These values are entered conversationally at the start of each new run.

The variables that may be altered for the second and subsequent runs are

(1) Reflux ratio
(2) Distillate withdrawal rate
(3) Feed condition
(4) Feed tray position

Any number of runs may be attempted, providing the C.P.U. time limit for the execution is not exceeded, but only five detailed result tables are allowed (see section 5.33). The limit is imposed to encourage users to determine whether or not a detailed result table is warranted by analysing the abbreviated result table printed on the terminal, and to prevent the generation of excessive amounts of computer printout.

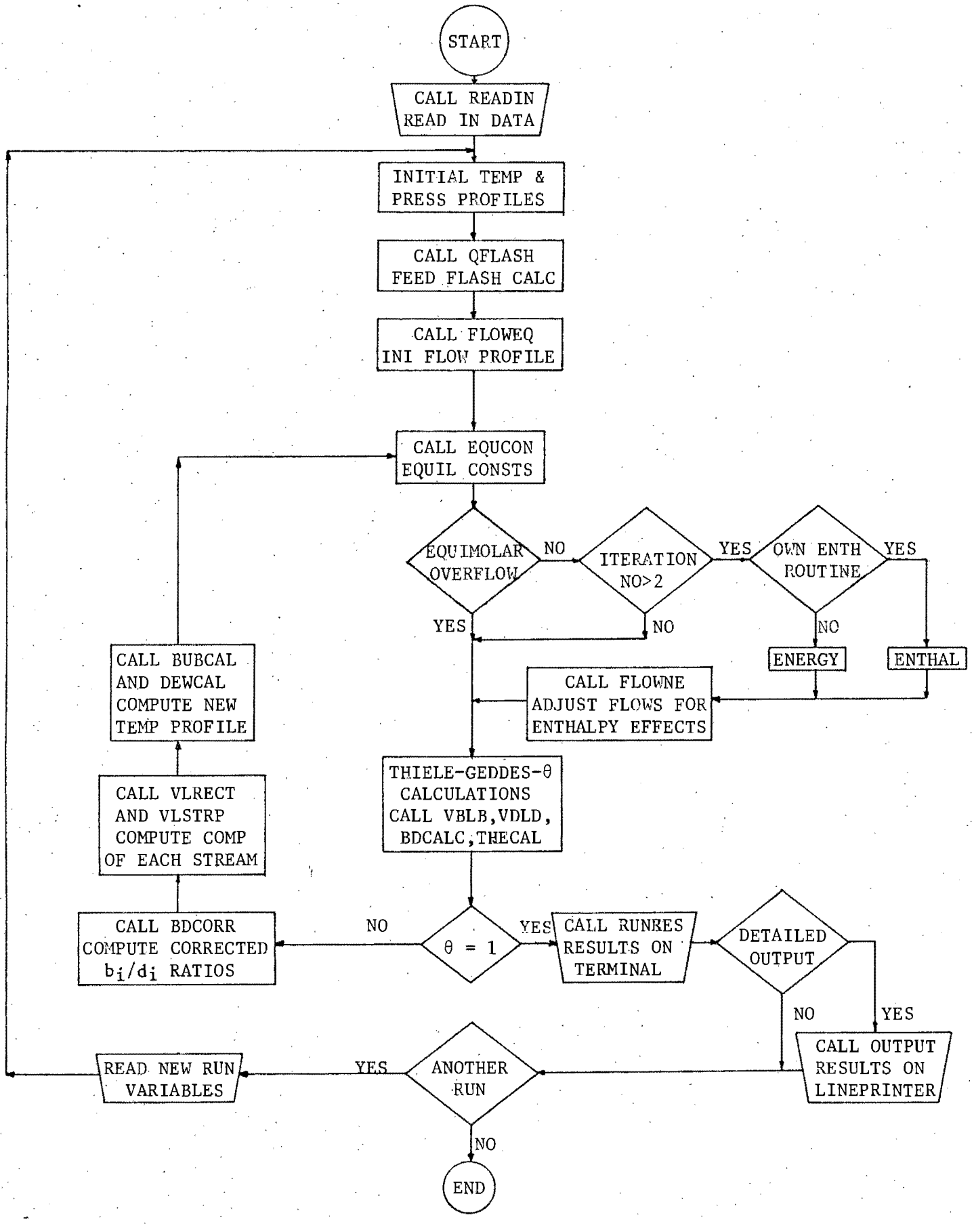


FIGURE 5.3 FLOWCHART OF PROGRAM MULCOM

5.33 Printout of results

Output of results is available in three forms:

(1) Abbreviated result table for primary analysis of a run. See subroutine RUNRES.

(2) Detailed result table showing composition and component flowrate data for the streams leaving each tray. See subroutine OUTPUT.

(3) Tables of results for use by the plotting program in constructing the following diagrams:

(a) Temperature profile.
(b) Liquid and vapour flowrate profiles.
(c) Distillate compositions in the form of a bar diagram.
(d) Composition profiles through the column.

Output (1) is written directly to the terminal after every run. It includes all the results commonly required in tutorial problems and permits an overall analysis of a system. The user is able to select variables for subsequent runs on the basis of these results, and may request a detailed printout (Output (2) above) if this is required.

Output (2) is optional and should only be requested if detailed composition and flowrate data are required. These data would be required for column sizing or tray design calculations. An output file is assigned to store the data before the program is executed. On termination of the program the file is directed to the high-speed line printer, or may be deleted if not required.

Output (3) is activated at the same time as output (2), but the data are stored only if the user has assigned the necessary files (see User Manual, section 7.435). The plotting routine superimposes the temperature profiles for

all the runs on one diagram for comparative purposes, but will plot diagrams (b), (c) and (d) for only one run at a time. (See User Manual, section 7.53). The user selects the run for which diagrams (b), (c) and (d) would be most useful, on the basis of outputs (1) and (2).

5.34 Termination procedure and error messages

When the execution is terminated the mainline routine will delete the file initially assigned for output purposes if it has not been used. If detailed result tables have been requested the mainline will inform the user of the commands necessary for sending the contents of the file to the line printer. The table is too lengthy for printing on the terminal.

Error messages are included to inform the user of any errors detected during the input of data. More detailed error messages are included in specific subroutines. Errors detected by the mainline routine result in termination of the program and deletion of the output file.

5.35 Limitations of MULCOM

The program is limited to the simulation of simple columns with known configurations, processing ideal multi-component solutions. Dimension parameters may be adjusted to accommodate the number of components and trays in a system. These are set at present for a maximum of 30 components and 50 trays (see MC,MP in COMMON block MULTICOM), although MC=10 and MP=20 would be sufficient for most textbook tutorial problems. Computer core storage limitations must be considered when setting these parameters (see User Manual, section 7.47).

5.36 Description of Fortran text

The mainline program MULCOM directs the calculation procedure as follows:

Line No.*	Program Action
6-8	Set initial values of control counters.
9-24	Start of program.
25	Call subroutine READIN to read all data required for the first run.
27-36	Arrays TRAYNO and COMPNO are used by the plotting routines. They contain a list of tray numbers and component numbers respectively.
37	Control is returned to this point at the end of a run.
38-45	Start of run and run identification.
48-70	New values of variables that may be altered for second and subsequent runs.
74-77	Calculation of initial temperature profile.
78-85	Calculation of tray-pressure profile.
88	Subroutine QFLASH called to determine feed condition.
90	Subroutine FLOWEQ called to compute initial flows in each section of column, assuming constant molal overflow.
92	Start of iterative calculation procedure (Thiele-Geddes algorithm).
94	Call subroutine EQUCON to determine equilibrium constants for each component on each tray.

*See FORTRAN text, Appendix C.

99-105 Enthalpy calculations through either ENERGY or ENTHAL if constant molal overflow conditions are not assumed.

106 FLOWNE computes flows leaving all trays when enthalpy effects are considered.

109 ABSTRP computes absorption and stripping factors for each component on each tray in the rectifying and stripping sections respectively (Equations (5.18 & 5.22)).

110,111 Subroutines VBLB and VDLD solve the stripping and reactifying sections respectively, working towards the feed tray. (VBLB - Equations (5.27, 5,28 and 5.29); VDLD - Equations (5.24, 5,25 and 5.26).)

112 Subroutine BDCALC matches the two sections at the feed tray (Equation (5.34)).

113 Equation (5.41) equality checked.

114 Subroutine THETA applies the Newton iterative method to determine a value of θ in Holland's θ convergence technique (Equations (5.50, 5.51 and 5.52).)

119,120 Convergence is achieved if $\theta \rightarrow 1,0$ or Equation (5.41) holds within accuracy limit ACCLIM. This terminates the iterative process.

125 Subroutine BDCORR corrects the b_i/d_i ratios using the calculated θ value. (Equations (5.45, 5.46 and 5.47).)

130,131 VLSTRP and VLRECT compute compositions of all streams in the stripping and rectifying sections respectively. (Equations (5.37 and 5.38).)

138-150 Subroutines DEWCAL and BUBCAL compute dew and bubble points on each tray, based on new composition values. These temperatures are averaged for each tray to yield a new temperature profile for the next iteration. (See Chapter 8.)

153 A maximum of 30 iterations is permitted to achieve convergence.

158 Subroutine RUNRES prints an abbreviated result table on the terminal. On the basis of these results a full set, detailing flows and compositions for each tray, may be requested.

167 OUTPUT writes a detailed set of results for the run to a previously assigned file, for printing on the line printer after execution is terminated.

168-195 Results for plotting of temperature, flowrate and composition profiles are written to file.

196 Return to line 36 above. If a further run is not required the program will then terminate.

197-204 Error messages are printed out if data are not supplied correctly. The program is then terminated.

206-217 Termination procedure. The user is told how to send the detailed result table to the line printer If this facility is not required, the unwanted file is deleted through FINISH.

5.4 GENERAL COMMENTS ON MULCOM

The program MULCOM has been designed as a teaching program and to this end many of the calculation techniques are highly indealized. A modular structure was adopted to facilitate incorporation of more sophisticated procedures

should these be required. The options included in the package should be sufficiently versatile to handle most standard tutorial problems.

The result table printed on the terminal in demand mode executions uses the standard tray numbering convention from the bottom of the column and provides data in SI units. In the internal numbering system the reboiler is stage 1 and not 0 to avoid using the zero address in arrays. The user is not normally concerned with this feature, however. Data supplied to the program may be in any convenient system of units providing that the conversion factors in the various subroutines are adjusted accordingly. These aspects are described in the User Manual (section 7.436).

Error messages are included in a number of subroutines. If negative flowrates are detected in the column the user is informed and corrective action is suggested (see subroutines FLOWEQ and FLOWNE). It is assumed that the user will have sufficient knowledge of distillation systems to understand the cause of such errors.

The application of the program to tutorial problems is discussed in Chapter 8. A number of forcing techniques, used to improve the stability of the iterative calculations and increase the rate of convergence, are discussed as they appear in the subroutines.

5.5 SUBROUTINES ASSOCIATED WITH MULCOM

All major input, output and computational operations within MULCOM are carried out by subroutines. Subroutines TEACH and SEARCH have been discussed in Chapter 2.

5.51 Subroutine READIN

This subroutine reads all the data required to solve a problem, excluding data required by a user's enthalpy

subroutine (see section 7.45). The formal statements in the FORTRAN text are self-explanatory and only intermediate calculations are described below.

Line no.	Program Action
4-11	DATA statements required for assigning files used to store plotting data. (See lines 196-210 below.)
16	Conversion factors for pressure units.
28-31	Feed tray position and condenser stage number are adjusted to internal numbering system. (See User Manual.)
33-63	Condenser pressure, tray pressure drop, condenser pressure drop, and condenser and reboiler temperatures are read in and the units converted to mm Hg and ^{o}C respectively.
96-103	Condenser type noted and counter NCOND set accordingly. NCOND = 1 - total condenser NCOND = 2 - partial condenser.
105-125	If the user is to provide an enthalpy routine this must be set up as described in the User Manual. Subroutine ENTHAL is an example of such a routine. Subroutine ENERGY includes three different enthalpy functions for which coefficients may be supplied. The required function is indicated by counter NENTH. The necessary coefficients are entered here for ENERGY whereas ENTHAL reads its own data as set out by the user.
126-144	Equilibrium data may be provided in any one of three forms. The function to be used is indicated by counter NEQUIL which is entered first,

followed by the appropriate data.

150 If a θ convergence limit is not specified by the user a value of 0,00005 is adopted.

163-177 Vaporization efficiency data may be supplied in two ways.

MEFF = 1 efficiency factors for each component on each tray required;

MEFF = 2 efficiency factors for each component on one tray required.

These are assumed to be the same for all other trays.

178-183 If perfect trays are assumed the efficiency factors are all set to 1,0.

196-211 Automatic assigning of files if plotting is required. File names must be 6 characters long and must be entered as:

AFILE1,AFILE2,AFILE3,AFILE4 (entered on one line)

5.52 Subroutine EQUCON

This subroutine computes equilibrium constants, $K_{i,j}$, and $K_{i,F}$ using one of four possible functions.

(1) Antoine's equation and $K_{i,j} = P_{i,j}/P_T$ (Equation (2.7))

(2) $K_{i,j} = \exp(a_i - (b_i/T_j) + c_i T_j)$ T_j in ^{o}R [18]

(3) $K_{i,j} = \alpha_{i-b,j} \cdot K_{b,j}$

where $\alpha_{i-b,j} = a_i + b_i T_j + c_i T_j^2$

and $K_{b,j} = c_i + e_i T_j + f_i T_j^2 + g_i T_j^3$ T_j in ^{o}F [21]

(4) $K_{i,j} = T_j[a_i + b_i T_j + c_i T_j^2 + d_i T_j^3]^3$ T_j in ^{o}R [21]

Tray efficiency is accounted for by multiplying the equilibrium constants by the vaporization efficiency factors $E_{V_{i,j}}$ as described in section 5.264. The feed stream equilibrium constants $K_{i,F}$ are calculated for use in the flash calculation when the subroutine is called by subroutine QFLASH. The feed flash equilibrium is assumed to be ideal. The procedure for selecting one of the above expressions and providing the necessary data is discussed in the User Manual (section 7.433).

Line no.	Program Action
3	Selection of equilibrium function.
5	NEQ = 1 when subroutine is called by QFLASH.
6-11	Calculation of $K_{i,j}$ using option (1) above.
12-17	'DO' loop 13 calculates $K_{i,F}$ values using option (1).
18-34	Similar calculation of $K_{i,j}$ and $K_{i,F}$ for option (2).
(21)	Tray temperature converted from ^{o}C to ^{o}R.
(22)	T_j in the above correlations represents the tray temperature x 10^{-2} for (2) and (3), and tray temperature x 10^{-3} for (4). This improves the significance of the coefficient when working in single precision. (See User Manual, section 7.433.)
34-50	Calculations using option (3).
52-67	Calculations using option (4).

5.53 Subroutine ENERGY

This subroutine is called either by the mainline program to calculate stream enthalpies and the condenser and reboiler heat duties, or by QFLASH to determine the feed stream enthalpy. Component enthalpies are computed using one of three functions of temperature. For the vapour phase:

(a) $H_i = (a_i + b_i T_j + c_i T_j^2)^2$ T_j in ^{o}R [21]

(b) $H_i = a_i + b_i T_j + c_i T_j^2$ T_j in ^{o}F [21]

(c) $H_i = a_i + b_i T_j + c_i T_j^2$ T_j in ^{o}R [18]

Similar equations define the liquid phase component enthalpy h_i. Coefficients for all three functions must be supplied for T_j = tray temperature x 10^{-2}, as discussed in section 5.52. Stream enthalpies are computed using Equations (5.4, 5.7 and 5.8).

Line no.	Program Action
3	Conversion factors (FPS to SI).
4,7	TRANK and POWER are defined by the choice of enthalpy correlation.

NENTH	TRANK	POWER
1	459,4	2
2	0,0	1
3	459,4	1

8,9	NEN is set by QFLASH.
10-23	'DO' loops 10 and 11 compute enthalpies of the vapour and liquid streams leaving each try in the column.
24-31	Condenser duty calculations (Equation (5.6)).

32-42	Calculation of feed stream enthalpy (Equation (5.4)). This calculation is required by QFLASH for NEN = 2 as well as for normal enthalpy balance purposes.
45	Reboiler heat duty by overall enthalpy balance (Equation (5.3)).
46,47	DM and BM are used in FLOWNE for determining flowrate profiles.
48-49	Unit conversion.
52-70	Calculations required only by QFLASH for NEN = 1. Computes enthalpy values for use in Equation (5.60).

Subroutine ENTHAL performs the same calculations shown above using the enthalpy correlation of Kobe [55] and the calculation technique described in Chapter 3 (subroutine HTCAP). ENTHAL is described fully in the User Manual, section 7.45.

5.54 Subroutine QFLASH

This subroutine calculates the q-value and phase compositions of the feed to a distillation column by flash calculation. The flash operation may be either isobaric or isenthalpic, with counter NQVAL recording the user's choice.

The routine computes bubble point and dew point values by calling subroutines BUBCAL and DEWCAL respectively. Subroutine EQUCON is called to compute equilibrium constants for the flash calculations and either ENERGY or ENTHAL is called to calculate enthalpy values.

The sequence of calculations is as follows.

(1) The bubble and dew points of the feed stream are calculated at the feed tray pressure.

(2) NQVAL, specified in READIN, selects calculation procedure for case (1) or (2) as described in section 5.263.

(a) Case (1):

If feed is superheated or supercooled the routine calculates the saturated liquid and saturated vapour enthalpy values and computes q according to Equation (5.60). For two phase feeds, an iterative flash calculation is performed, using Equation (5.59) above, to evaluate q and the composition of each phase.

(b) Case (2)

The same procedure used for isobaric systems is employed but the calculation techniques are as described in section 5.263. Error messages are included to prevent excessive looping of the iteration procedure.

(3) On completion of the calculations the computed value of q, the bubble and dew point temperatures and the actual feed temperature are printed out for examination by the user.

Line no.	Program Action
8-13	'DO' loop 10 sets initial values of $x_{i,F}$ and $y_{i,F}$ and stores these in arrays X and Y for use in subroutine BUBCAL and DEWCAL. Address IT is unused in the calculation procedure and it is for this purpose that the tray-dimension parameter MP must be equal to at least (NP+3) (see User Manual).
16-18	In BUBCAL and DEWCAL, MD = 2 indicates that a temperature is required for "tray" IT only. The bubble point and dew point for a stream of composition Z_i are computed.
19	NQVAL = 1 isobaric flash calculation. NQVAL = 2 isenthalpic flash calculation.

(1) Isobaric flash

21 If feed is superheated or supercooled calculations continue at line 50.

22-45 Iterative flash calculation using Equation (5.59) (line 30).

46-47 Moles of vapour and liquid in feed.

50 If enthalpy data are not supplied the feed q-value must be entered directly.

51-73 Determination of feed q-value using Equation (5.60) for superheated or supercooled feeds.

76-83 Feed q-value requested if enthalpy data are not available and feed is superheated or supercooled.

84 Return to line 56 to set feed compositions and flowrates.

(2) Isenthalpic flash

86-91 Check for superheated or supercooled feed.

92-135 Iterative search for V_F:L_F ratio that results in $\Sigma x_{i,F} = 1,0$ with the calculated H_F equal to the specified feed enthalpy FENTH. Equation (5.58) is used to calculate the $x_{i,F}$'s.

137-164 Determination of temperature of superheated or supercooled feed such that the calculated H_F equals the specified enthalpy FENTH.

165-169 If the isenthalpic flash calculation for two phase feeds fails, an error message is printed out and the execution is terminated.

172-176 Feed q-value and temperature data printed out.

5.55 Subroutines BUBCAL and DEWCAL

The use of Newton's convergence method in these subroutines is similar to that described for the binary system routines in section 2.55. The two routines are different only in the form of the functions used and thus the following description of the calculation procedure applies to both. The four functions used to determine equilibrium constants must be used to compute corresponding bubble points and dew points. The function and its first derivative are required in Newton's method. Efficiency factors are included to account for pseudo-equilibrium conditions as discussed in section 5.264. Allowance is made for the assumption that a total condenser acts as perfect stage irrespective of the specified efficiency data [21].

An important feature of the routines BUBCAL and DEWCAL is the inclusion of an averaging procedure. It was found by Lyster et al. [33] that the combination of the Thiele-Geddes calculation procedure and the θ method of convergence could lead to overcorrection of the independent variables from one iteration to the next.

In the case of temperature profiles Lyster observed that by averaging the new and old temperature values on each tray and adopting the mean as the correct value for the next iteration, the convergence characteristics were enhanced. In this study both the bubble point and dew point are calculated for each tray and averaged with the previous value as suggested by Lyster. The final temperature profile for the next iteration is then taken as the mean of the bubble and dew point values on each tray (lines 148 in mainline program). This is discussed in detail in Chapter 8.

The calculation procedure shown below applies to both routines.

Line no.	Program Action
3	Choice of equilibrium correlation as in EQUCON.
4-29	Calculation using Antoine's equation.
6-21	Newton's iterative convergence method.
(11)	MMD > 1 when called by QFLASH for IT = NP2+1 or when total condenser is considered a perfect stage. In both cases efficiency is not considered and EFF = 1,0.
(12,13)	Antoine's equation ($f(T) = P_T - \Sigma P_{i,j} E_{V_{i,j}} x_{i,j}$)
(14,15)	$F'(T) = -\Sigma x_{i,j} . E_{V_{i,j}} . P_{i,j} . BSQ$, hence TOLD = TOLD + DIV.
23-29	Averaging procedure.
30-37	Calculations using option (2) in EQUCON. The procedure is identical, with a different function used.
39	$f(T) = \Sigma E_{V_{i.j}} . K_{i,j} x_{i,j} - 1.$
41	$f'(T) = \Sigma x_{i,j} . DALFA.$
59-86	Calculations using option (3) in EQUCON.
73	$f(T) = E_{V_{i,j}} K_b - \frac{1}{\Sigma \alpha_{i,j} x_{i,j}}$
74	$f'(T) = E_{V_{i,j}} \frac{dK_b}{dt} + \frac{\Sigma \alpha_{i,j} \frac{d\alpha_{i,j}}{dT}}{(\Sigma \alpha_{i,j} x_{i,j})^2}$
87-115	Calculation using option (4) in EQUCON.
103	$f(T) = \Sigma E_{V_{i,j}} . K_{i,j} . x_{i,j} - 1.$
104	$f'(T) = \Sigma x_{i,j} . E_{V_{i,j}} \frac{dK_{i,j}}{dT}$

Temperatures are converted back to ^{o}C before return to the calling program in option 2, 3 and 4 above. Antoine's equations use temperature in ^{o}C.

5.56 Vapour and Liquid flowrate profiles

5.561 Subroutine FLOWEQ: This subroutine computes the initial flowrate profiles through the column, assuming constant molal overflow. These profiles are maintained for two iterations when enthalpy effects are considered, to allow a degree of stability to be achieved. This procedure was suggested by Holland [21], who found that introducing enthalpy effects immediately could cause the convergence to fail for some systems.

Error messages are included to indicate any detected negative flows. As these are usually the result of a low reflux ratio specification the user is permitted to enter an increased value. To terminate the execution, a reflux ratio of 1 000 or greater is entered.

The standard material balance equations described in Chapter 3 for McCabe-Thiele assumptions are used in the routine. Comment cards in the FORTRAN text indicate the calculation sequence.

5.562 Subroutine FLOWNE: This subroutine is used to adjust the vapour and liquid flowrates according to the enthalpy balances. It is called by MULCOM only when enthalpy effects are considered. The conventional method for performing enthalpy balances is used in the routine [21].

The vapour and liquid flowrates leaving the trays in each section of the column are determined by alternate application of the material and enthalpy balance equations for each section.

An important feature of this subroutine is the "forcing" procedure that is included to restrict the magnitude of change in flowrate profiles, between successive iterations. This forcing technique was first proposed by Lyster et al.

[33], to counteract the instabilities caused by overcorrection of variables from one iteration to the next. It is similar in operation to the averaging procedure included in BUBCAL and DEWCAL.

Lyster observed that, when enthalpy balances were included, the deviation of the calculated value of D, from the specified value, was amplified. This appeared to be partly due to the fact that both the temperature and the flowrate profiles were changing with successive iterations, resulting in a general increase in perturbation within the system. This effect increased with increase in boiling point range of the feed components, to the point of causing instabilities and failure of the convergence procedure. Since the assumption of constant molal overflow enabled all the problems to be solved, Lyster reasoned that stability could be achieved by limiting the change in flowrate profiles for successive iterations.

The following procedure was proposed by Lyster, and has been adopted in this subroutine.

(a) The first two iterations are carried out assuming constant molal overflow conditions. The purpose of this step is to allow the temperature profile to adjust partially, before additional disturbance is imposed on the system.

(b) The maximum and minimum value of each new flowrate calculated in FLOWNE is limited to 1,2 and 1/1,2 times the previous value for the third, fourth, fifth, sixth and seventh iterations.

(c) These limits are reduced to 1,1 and 1/1,1, and 1,025 and 1/1,025 in the eighth and thirteenth cycle respectively.

(d) If the calculated flowrate falls within the limits imposed for any trial, the new value is averaged with the previous value to obtain the correct profile.

As most simple tutorial problems involve feeds with narrow boiling point ranges, this forcing procedure will often be unnecessary. It has, however, proved useful in stabilizing the solution of certain systems, as will be discussed in Chapter 8.

An error message indicates the occurrence of any negative flowrates. The run is then terminated, after the position of the calculated negative flow in the column has been printed out. Such errors are usually the result of low reflux ratios coupled with low feed q-values.

Line no.	Program Action
3-5	Iteration-number counter determines "forcing procedure" limit.
6-20	'DO' loop 11 computes flowrates leaving tray in the rectifying section.
(7-8)	Previous value stored and limits of new value calculated.
(10)	Enthalpy balance determines liquid flowrate (modified Equation (5.6) with DM computed in subroutine ENERGY or ENTHAL).
(12-18)	New value restricted within calculated limits.
(19)	Material balance around condenser and tray n.
21-35	'DO' loop 21 computes flowrates in stripping section. The calculation sequence is the same as for the rectifying section.
(25)	Enthalpy balance (modified Equation (5.10) with BM from ENERGY or ENTHAL).
(34)	Material balance around tray m and reboiler.
36-41	Flows leaving ends of each section of column are set. For a partial condenser the distillate

product is a vapour (line 39).

42-59 Error messages indicate detected negative flowrates and terminates the program.

5.57 Thiele-Geddes tray-to-tray calculations.

The Thiele-Geddes algorithm has been divided into the following sections, each of which is contained in a separate subroutine.

(a) Calculation of absorption and stripping factors: Subroutine ABSTRP.

(b) Calculation of $v_{i,m}/b_i$ and $l_{i,m}/b_i$ ratios in the stripping section of the column: Subroutine VBLR.

(c) Calculation of $v_{i,n}/d_i$ and $l_{i,n}/d_i$ ratios in the rectifying section of the column: Subroutine VDLD.

(d) Calculation of b_i/d_i ratios at feed tray position: Subroutine BDCALC.

(e) Calculation of the "corrected" b_i/d_i ratios using the computed value of θ: Subroutine BDCORR.

(f) Calculation of component molar flowrates, total molar flowrates and component compositions of both phases on each tray in the stripping and rectifying sections: Subroutines VLSTRP - stripping section, and VLRECT - rectifying section.

The subroutines are called by the mainline in the order shown above. Subroutine THECAL is called between BDCALC and BDCORR to determine θ, as described in the following section. Comment cards on the FORTRAN texts indicate the calculation sequence where necessary.

Subroutines ABSTRP, VLRECT and VLSTRP include a COMMON block whereas the other four use calling lists. FORTRAN

equations in the subroutines relate to the equations shown in section 5.2 as follows. Counter NCOND defines the type of condenser specified, which sets the starting values for calculations in the rectifying section.

Line no.	Calculation performed
Subroutine ABSTRP:	
5	Equation (5.20)
10	" (5.19)
15	" (5.18) including (5.31) (Flowrate values set in FLOWEQ or FLOWNE)
20	Equation (5.22) and (5.23)
Subroutine VBLB:	
5	" (5.27)
9	" (5.29)
10	" (5.28)
Subroutine VDLD:	
5	" (5.24)
9	" (5.25)
10	" (5.26)
Subroutine BDCALC:	
6	" (5.34)
7	" (5.35)
8	" (5.41) (yields D_{calc})
Subroutine BDCORR:	
6	" (5.45)
7	" (5.46)
8	" (5.47)
9	" (5.48)

Line no.	Calculation performed
Subroutine VLRECT:	
5	$v_{i,n} = (v_{i,n}/d_i)d_i$
6,8	$V_n = \Sigma v_{i,n}$
11,18	Equation (5.38)
20	Equation (5.5) (for partial condenser)
24	Composition normalization
39	Equation (5.39)
40	" (5.40)
Subroutine VLSTRP:	
7	$v_{i,m} = (v_{i,m}/b_i)b_i$
12,13	Equation (5.44) (5.43)
15	" (5.37)
16	" (5.38)

At the end of these calculations the vapour and liquid compositions on each tray are known. These data are then used in BUBCAL and DEWCAL to determine a new temperature profile.

5.58 Subroutine THECAL

The theory of Holland's θ method has been discussed previously. The application of the θ method in this program is discussed here. Two convergence criteria are tested in the mainline program on return from THECAL. They are:

(1) $|\theta - 1,0| \leq$ ACCLIM

(2) $|D_{calc} - D_{spec}| \leq$ ACCLIM and $|\theta - 1,0| \leq 0,001$ where D_{calc} is computed in BDCALC and ACCLIM is the limit of accuracy, which is either specified by the user (see section 5.51) or taken as 0,00005. The value 0,00005 is suggested

by Petryschuk et al. [31] for computation in single-precision. The above two criteria have been used most often by workers investigating the Thiele-Geddes-θ solution procedure [21,31, 33,35] and although it is customary to consider only one criterion, both have been included in this application for the following reasons.

(a) As observed by Naphtali [27], it was found that for some systems the rate of convergence decreased as the solution was approached. The value of θ converged very slowly or, in some cases, started to oscillate about a value close to 1,0 but not within the specified limit of accuracy. It was found however, that in most of these examples a satisfactory solution could be obtained by applying criterion (2) shown above. Allowing the program to run until criterion (1) was also met, in examples where θ did not oscillate, produced no discernible change in the final results, and thus the extra computer time required was not justified. Systems exhibiting these characteristics usually involved (i) feeds with wide boiling point ranges, resulting in end-product composition of some components tending to zero, or (ii) columns with pinch zones extending across several trays. (See Chapter 8.)

(b) Multicomponent systems normally considered in tutorials involve a narrow boiling range feed and a column with little or no pinch zone. For systems such as these the θ criterion was always met first.

Thus the use of two convergence checks ensured that a problem would be solved in the shortest possible time, if convergence could be achieved at all.

THECAL is called by the mainline routine after the b_i/d_i ratios for the particular iteration have been computed in BDCALC. The purpose of the routine is to compute a value of θ such that Equation (5.49) holds.

Newton's iteration procedure is employed, using Equations (5.50) to (5.52), as discussed above. The previous value of θ is taken as the starting point for the new calculation each time THECAL is called, except in the following instances:

(i) For the first iteration θ has not been previously calculated, and a starting value of $\theta = 0{,}1$ is used.

(ii) If the previous value of θ is greater than 9999, the initial value is taken as 0,1.

If θ becomes negative during the iterative procedure, it is multiplied by -1 and the calculations are continued. If the absolute value of θ exceeds 10 000 at any stage during the iterative procedure, control is returned to the mainline where θ is set equal to 10 000. This is done to avoid unnecessary computation for a system that is far from a converged solution. Extremely large values of θ were found to be associated with the type of system described in (a) above. In such cases it was found that by constraining θ as described above, the calculation procedure stabilized rapidly, and converged within ten iterations of θ first being adjusted, with criterion (2) usually applying.

Line no.	Program Action
3	Check limits of θ (THETA).
4,5	Starting value $\theta = 0{,}1$
10	Equation (5.51).
11	Equation (5.52).
15-18	Equation (5.50).
19-21	Limits of θ allowed: $0 < \theta < 10\,000$
24	Convergence limit on Newton's iterative procedure.

5.59 Output Procedures

The three types of output available from MULCOM have been described previously. Data for graphical representations are written to file by the mainline program. Two subroutines are used to point result tables for analysis by the user.

5.591 Subroutine RUNRES: This subroutine is designed to provide a compact but comprehensive result table on the terminal, during demand mode executions.

The results printed out by RUNRES include:

(1) Overall material balance.

(2) Compositions of distillate, bottoms and feed streams. The feed stream data include the overall values entered by the user and the liquid and vapour phase composition computed by flash calculation.

(3) Tray Temperature and Pressure profiles. The temperature profile is shown in both ^{o}C and ^{o}F for easy comparison with solutions given in the literature.

(4) Number of iterations required for covergence and final value of θ.

(5) Reflux ratio, distillate rate.

(6) Feed conditions - Q value from flash calculation
- Feed temperature
- Feed enthalpy

(9) Condenser and reboiler heat duties.

(10) Type of condenser employed - partial or total.

(11) Flowrates of vapour and liquid leaving each tray.

Sample printouts are shown in Chapter 8. The format statements in the FORTRAN text are self explanatory and will not be discussed here. The internal tray numbering system is converted to the conventional system through implied DO loops as the data are printed out. If complete composition

or molar flowrate profiles are required, the detailed result table must be called for.

The printout format is designed for terminals with a restricted carriage length of 70 characters. The detailed printout, discussed next, is designed for a full 130 character line printer output.

5.592 Subroutine OUTPUT: This subroutine produces a detailed table showing compositions and molar flowrates of streams leaving each tray in the column. These data would not be required for standard tutorial solutions, unless an analysis of the composition profiles is necessary. Detailed tray data are normally required in tray design calculation or for column sizing purposes. A sample printout is shown in Chapter 8 and on the UNIVAC system this is produced on the high-speed line printer.

The WRITE key NLP is set by the mainline routine. The file corresponding to this key is assigned to the run by the user before execution is initiated. The key can be adjusted by the user to refer to any convenient output device. The printout format assumes that the printer has a carriage width of at least 120 characters.

5.6 SUMMARY OF MULTICOMPONENT SIMULATION PROGRAM

Program name: MULCOM
Type: Mainline with subroutines.
Operation Mode: Demand or Batch on UNIVAC 1106.

1. Title: Simulation of multicomponent distillation systems.

2. Solution type: Rating program.

3. Application: MULCOM uses the Thiele-Geddes calculation procedure to simulate distillation operations in columns of known configuration. Holland's θ method is used to improve the rate of convergence of the iterative calculations.

The program will consider simple columns producing a single phase distillate from either a partial or a total condenser. A partial reboiler is assumed with a liquid Bottoms product withdrawn. Ideal solution behaviour is assumed and a choice of four equilibrium constant correlations is included. Enthalpy effects may be considered if data are available. Provision is made for including enthalpy correlations of arbitrary complexity. Stage efficiency may be considered in terms of component vaporization efficiency factors.

The user may repeat a run using different values of certain variables without terminating the execution. In demand mode an abbreviated result table is printed on the terminal at the end of a run. A detailed result table may be directed to the line printer if required and a plotting program is available for graphical representation of temperature, flowrate and composition data.

4. Program Development:

a) Computer used: UNIVAC 1106 - EXEC 8 control.
b) Minimum core requirements: 17 K
c) Language: Standard FORTRAN IV.
d) Typical run time: 5 secs

CHAPTER 6

GRAPHICAL REPRESENTATION OF RESULTS

6.1 INTRODUCTION

The plotting routines included in this computer program package have been developed specifically to supplement the output from each program, excluding the packed column design routine PACKED.

The plotting software published by CALCOMP* must be available, together with an integrated hardware plotting device.

The plotting routines are linked to a mainline program that operates conversationally, in demand mode. The mainline program and the associated subroutines call the standard 'CALCOMP' routines as required. A number of data elements are used to store the diagram axis titles. These are added to the runstream through the ERTRAN facility and can be changed by the user if necessary.

Data files are assigned by the user during the execution of a particular distillation program. These files are added to the plotting runstream as required through the ERTRAN facility.

The following diagrams may be called for by the user:

A. Multicomponent system.
   (1) Temperature profile through the column.
   (2) Vapour and liquid flowrate profiles through the column.
   (3) Distillate composition bar diagram.
   (4) Composition profile through the column.

B. Binary system.
   (1) McCabe-Thiele diagrams
      (a) Batch distillation - Variable reflux ratio
      (b) Batch distillation - Constant reflux ratio
      (c) Continuous distillation.

* CALCOMP - CALIFORNIA COMPUTER PRODUCTS INC.

(2) Ponchon-Savarit diagrams

(a) Continuous distillation - tray columns.

6.2 MAINLINE PROGRAM FOR PLOTTING ROUTINES - PLOTTER

PLOTTER serves to link the subroutines together and to write the identifying titles above each completed diagram. It initializes each plot by calling the standard CALCOMP routine PLOTS. Thereafter the relevant subroutine is called to construct the specified diagram.

The program is divided into three sections. The first deals with multicomponent systems, the second with binary batch systems and the last with binary continuous systems. These will be considered in order.

The operating runstream is fully discussed in the user manual, section 7.5. A description of the internal program organization is given here.

Line No.*	Program Action
23-28	Requests run identification number and date for reference purposes. This number is written at the top of the completed diagram (see lines 155-159).
29-31	Requests type of plot required. Code numbers 1-5 identify specific diagrams (see User Manual).
32	Assigned GO TO statement selects program section.
(a) Multicomponent systems.	
34-57	Input of execution data (see User Manual for details).
58-76	Add title data elements and read in diagram headings and axis titles (see User Manual).
77	Plotting initialized through CALCOMP subroutine PLOTS.

*See FORTRAN text, Appendix C.

78	CALCOMP subroutine FACTOR scales overall size of diagram.
79-82	Calling of plotting subroutines for construction of each diagram.
83-93, 155-159	Identifying heading and job number written above diagrams, using CALCOMP subroutines SYMBOL and NUMBER.

(b) Binary Batch systems.

95,96	Plot initialized and scaled.
97	Subroutine PLOTBT constructs McCabe-Thiele diagram for batch system.
99	Diagram titles for constant reflux operation.
102	Diagram titles for varying reflux operation.
104-116	Titles read and written out.
(155-159)	Job identification number.

(c) Binary continuous systems.

118-122	Limits of enthalpy axis in enthalpy-concentration diagram (required for Ponchon-Savarit diagrams).
123-124	Plot initialized and scaled.
125	Subroutine PLOTMT constructs either McCabe-Thiele or Ponchon-Savarit diagrams as specified through counter NTYPE (see User Manual).
127-137, (155-159)	Read and write title and job identification number on Ponchon-Savarit enthalpy concentration diagram.
140-150, (155-159)	Read and write title and job identification number on McCabe-Thiele or Ponchon-Savarit x,y diagram.
160	Plot terminated through CALCOMP subroutine PLOT.

6.3 DIAGRAM-CONSTRUCTING SUBROUTINES

6.31 Subroutine PLOTTS

This subroutine plots the temperature profiles for each of the runs stored on file by MULCOM. They are all plotted on a single set of axes for easy comparison and each curve is identified through use of different plotting characters. The runs are plotted in the order they were stored on file.

Line No.	Program Action
19-28	The length of the x-axis (tray numbers) is determined, based on the total number of trays in the column. The maximum value is 8 inches. For columns with less than eight trays a scale of 1" per tray is used. From 8 to 16 trays, ½" per tray is used.
29	The length of the y-axis is supplied by the mainline from element AXISTITLE. This is presently set at 8" but can be changed by the user if necessary.
30	CALCOMP subroutine PLOT sets the origin of the axes.
31-34	The temperature data file is added through ERTRAN and temperature vs tray number data for the first run are read.
35-36	CALCOMP subroutine SCALE calculates the scale on each axis based on the defined length and the data to be plotted.
37-42	The appropriate axis titles are selected from the arrays XAX and YAX created by the mainline line.

43-44	CALCOMP subroutines MAXIS and AXIS set up and labelled X and Y axes respectively.
46	CALCOMP subroutine LINE plots data for first run.
49-59	'DO' loop 11 reads and plots data for remaining runs.
62-72	'DO' loop 20 and 21 select diagram titles from arrays ALXT, ALYT and TITL, provided by the mainline from element TITLES, and write these at the top of the diagram.
74-83	Plotting symbol legend written above diagram identifies each run's temperature profile.
84-87	The diagram is completed by a boundary line and control is returned to the mainline. ALX1 and ALY1 position pen correctly for next diagram.

6.32 Subroutine PLOTFS

The vapour and liquid flowrate profiles for a single run are plotted by this subroutine. The user specifies the data set to be plotted at the start of the run (see User Manual). Only one set is plotted because the diagram becomes difficult to read if too many lines are drawn. The user is able to extract data from the MULCOM printouts for comparison, and could insert these manually using different colours for clarity. The program selects the specified data set from the flowrate data file using counter NRF.

Line No.	Program Action
18-30	Axes are sized, origin set up and data file added through ERTRAN as described in section 6.31.
31-38	'DO' loop 10 selects the specified data set from the file.

39-52	Axes are scaled, set up and labelled as described in section 6.31.
53,54	Liquid and vapour profiles are plotted on the diagram.
56-66	Diagram headings selected and written.
67-77	Symbol legend identifying liquid and vapour profiles inserted.
78-82	Boundary line drawn and control returned to the mainline. ALX1 and ALY1 act as before.

6.33 Subroutine PLOTDS

This subroutine constructs a bar diagram indicating the component distribution in the distillate product. The specified run data are selected in the same way as the flowrate profile data, using counter NRF.

Line No.	Program Action
18-23	Axis lengths determined. X-axis (component number) is limited to 18 inches which is the maximum width of the full multicomponent system plot (see Chapter 8). The Y-axis is set at 5" with a scale of 0,2 mole fraction per inch.
24-30	'DO' loop 10 selects axis titles from XAX and YAX and PLOT sets origin of axes.
31-36	The data file is added through ERTRAN. 'DO' loop 11 selects specified data set.
37,38	Origin and interval of plotting on the Y-axis are set.
42,43	PLOT is used to draw X-axis.
45-52	'DO' loop 12 numbers the bar position. The width of a bar is 1" per component for systems of

	less than 18 components. For larger systems the 18" length is divided into the appropriate number of equal increments. The component numbers read in order of decreasing volatility with the lightest component numbered 1.
53-55	The X-axis title is written in.
56	AXIS sets up the Y-axis (mole fraction) in the normal way.
58-66	'DO' loop 13 draws in the bars of width DEL, using subroutine PLOT.

The remainder of the routine writes the titles above the diagram and constructs a boundary line as described previously (section 6.31 and 6.32).

6.34 Subroutine PLOTCS

This subroutine plots the composition profiles through the column for each component for the specified run.

Line No.	Program Action
19-29	PLOT sets the origin of the axes, ERTRAN adds the data file and 'DO' loop 10 and 11 search for the correct data set by reading through the file until the specified set is reached.
31-35	The data for the light component is then read and the y-axis start value and increment set.
36-49	Axis titles selected, X-axis scaled and axes set up using MAXIS and AXIS. The Y-axis uses the same length and scale as used in subroutine PLOTDS.
50	Data for light component plotted.
52-62	'DO' loop 13 reads and plots data for remaining components.

63-74	The size and format of the legend table identifying the profiles depends on the number of components. A maximum of 30 components is allowed for in three columns of 10 components each. The minimum size of a column is 5 components (see Chapter 8).
76-86	'DO' loops 20 and 21 select and write titles above diagram.
87-105	'DO' loops 22 and 23 write the legend table.

The diagram is then completed with a boundary line and control is returned to the mainline.

6.35 Subroutine PLOTBT

A McCabe-Thiele x,y diagram for binary batch distillation systems operating under either constant or varying reflux conditions is constructed by this subroutine. The plotting procedure is the same in each case and the type of operation is indicated by counter NTYPE (see User Manual). Element AXISMT contains the axis titles. This element and the data file BDPLOT are added through ERTRAN. The overall size of the diagram is set at 10" x 10". This can be scaled down using the CALCOMP subroutine FACTOR rather than by reducing the axis lengths.

Line No.	Program Action
9-10	Length, starting value and plotting increment are set for each axis.
11-16	AXISMT added and axis titles read.
17	PLOT sets origin of axes.
18-19	X and Y-axes set up and labelled using AXIS.
20-22	PLOT completes boundary line around diagram.
23-24	PLOT constructs 45^{o} diagonal on diagram.

25-32	Data file added, equilibrium data read and equilibrium curve constructed.
33-54	'DO' loops 20 and 21 perform the following for initial and final batch-run conditions.
(34-36)	Read in number of trays, coordinates of steps on x,y diagram and operating line coordinates.
(37-40)	Use PLOT to construct operating line.
(45)	Insert stepped stages on diagram.
(48-53)	'DO' loop 21 numbers the stages from the bottom of the column.

At this point the diagram is complete for variable reflux operations.

(56-72)	For constant reflux operations the average distillate product composition is inserted.

Control is then returned to the mainline.

6.36 Subroutine PLOTMT

This subroutine constructs either McCabe-Thiele or Ponchon-Savarit diagrams for binary continuous-distillation systems. The type of diagram required is indicated by counter NTYPE in the calling list.

The first section of the program constructs the enthalpy-concentration section of the Ponchon-Savarit diagram. Element AXISPS contains the axes titles for this section.

The size of the enthalpy-concentration diagram is set at 15 inches for the y-axis and 10 inches for the x-axis. PLOT is used to set the origin of the diagram such that the x,y diagram can be drawn below it, (see Chapter 8).

The enthalpy axis coordinates of the "Δ" points must be specified by the user (see User Manual). These are

extracted from the result table supplied by program BINARY. The values are requested by the mainline and are included in the calling list. The routine uses the values to determine the scale on the enthalpy axis. The lower value supplied is taken as the starting point for plotting enthalpy data. It is usually convenient to round the values off as this will result in a neater diagram with the Δ points lying inside the boundary line.

Line No.	Program Action
(a) Enthalpy-concentration diagram.	
16-23	Axis titles are read and the axis lengths set.
24	PLOT sets the origin of the axis to allow space for the x,y diagram below it.
30-31	Y-axis scaled using Δ values of enthalpy. Y-axis constructed and labelled using AXIS.
33	X-axis constructed and labelled.
34-51	Data file added through ERTRAN. Coordinates of points representing the feed, reboiler, condenser and Δ point are read. The lines joining these points are constructed using PLOT.
52-64	These points are labelled on the diagram.
66-73	'DO' loop 11 reads in the enthalpy curve data and plots this on the diagram. Liquid enthalpy is supplied first by the data file.
74-78	Boundary line construction completes diagram.
(b) x,y diagram	

The same basic procedure is used for both design methods. Where necessary variations are indicated by counter NTYPE. The size of the diagram is set at 10" x 10". Element AXISMT provides axis titles as described in section 6.35.

79-98	Axes set up and diagonal constructed as described in section 6.35.
100	Data file added here for McCabe-Thiele designs.
102-108	Equilibrium curve constructed as described in section 6.35.
110-121	Operating lines constructed on McCabe-Thiele x,y diagram. Complex columns allowed for by NQDAT.
124-133	'DO' loop 12 constructs the two operating lines on a Ponchon-Savarit x,y diagram.
134-137	Number of trays and coordinates of steps on x,y diagram read for either design method.
139-140	Point of intersection of operating lines must be read for Ponchon-Savarit diagrams.
143-146	Starting values and plotting increments set for both axes.
147	Equilibrium stages stepped off.
148-155	'DO' loop 21 numbers the stages from the bottom of the column (reboiler = 0) (see Chapter 8).
156-168	'DO' loop 30 labels feed and side stream positions and inserts 'q' lines.

Control is then returned to the mainline.

6.4 GENERAL COMMENTS ON THE PLOTTING FACILITY

Graphical representation of data is useful for comparing and analysing the effects of variables on a system. Constructing operating diagrams or composition profiles manually from tables of data is usually a tedious and time consuming process. The plotting facility developed here enables the user to obtain

graphical solutions to a number of problems in a short period of time. More time can thus be devoted to a critical analysis of the results. This aspect is discussed further in Chapter 8.

The use of data elements to store axis labels and other diagram titles enables the user to alter these without editing the program or subroutines. The size of each diagram is limited so as to prevent wastage of plotting paper, while ensuring that the diagrams are clear. CALCOMP subroutine FACTOR is used to scale the whole diagram to a convenient size such as A4. This procedure results in a better resolution than is obtained if the size of the original plot is reduced. The set of multicomponent plots is limited to a maximum size of 18" (x direction) by 28½" (y direction). The four graphs are arranged as shown in Chapter 8.

6.5 SUMMARY OF PLOTTING PROGRAM

Program name: PLOTTER
Type: Mainline with subroutines.
Operation mode: Demand or Batch.

1. Title: Plotting routines associated with the distillation teaching-aid program package.

2. Solution Type: Graphical representation of data.

3. Application: PLOTTER may be used to construct the following diagrams using data generated by programs in the distillation package.

(A) Multicomponent systems (program MULCOM)
  (1) Temperature profile
  (2) Vapour and liquid flowrate profiles
  (3) Distillate composition bar diagram
  (4) Composition profiles.

(B) Binary systems.
  (1) McCabe-Thiele diagram for batch operation (program BATCH).

(2) McCabe-Thiele diagram for continuous operation (program BINARY).

(3) Ponchon-Savarit diagram for continuous operation (program BINARY).

The program uses the CALCOMP software and hardware facilities available at the University of Cape Town. Similar facilities would be required to run this program on other computer systems.

4. Program Development:

a) Computer used UNIVAC 1106 with CALCOMP plotter

b) Software requirements: CALCOMP plotting software.

c) Minimum core requirements: 20 K

d) Language used: FORTRAN IV

e) Typical run time: CPU : 7 secs
Plotting: 4 mins (penplotter)

CHAPTER 7

DISTILLATION PACKAGE USER MANUAL

INTRODUCTION

This manual is designed to guide users in the operation of the distillation programs and associated plotting routines. Extracted from the main text together with Appendix A, the nomenclature tables and list of references, it provides all the necessary information for using the programs to solve standard tutorial problems. The options included in the programs should make them sufficiently versatile for undergraduate tutorial purposes. Where it might be necessary to change the FORTRAN text of a program for research or other purposes, the user is referred to the original thesis. In order to produce a self standing manual a certain amount of duplication of the main text was unavoidable.

The computer programs described here cover the following aspects of distillation taught in an undergraduate course at U.C.T.

(1) Binary batch distillation in tray towers - section 7.1.
(2) Binary continuous distillation in tray towers - section 7.2.
(3) Binary continuous distillation in packed towers - section 7.3.
(4) Multicomponent continuous distillation in tray towers - section 7.4.

A plotting program is available for graphical representation of results generated by programs 1, 2 and 4 above. (See section 7.5.)

A summary of the general operating procedures and conventions applying to all programs is given in section 7.6.

The computer executive commands listed in the following text apply to the UNIVAC 1106 - EXEC 8 computer system currently installed at U.C.T. It is assumed that the user has a working knowledge of the UNIVAC EXEC 8 control language and is able to execute programs using either a card deck (batch mode) or a computer terminal (demand mode). All distillation programs are contained in a file named *DISTFILE and all plotting routines in *PLOTFILE. The project name under which these files are stored on the system is set by the course supervisor. A master copy of both files should be kept on cards for longterm security, and on magnetic tape for normal operating purposes.

7.1 BINARY BATCH DISTILLATION

7.11 Application

The program BATCH simulates binary batch-distillation systems, operating under either variable or constant reflux conditions. All systems considered are subject to the following limitations:

(a) The number of trays in the column must be specified.
(b) Normal McCabe-Thiele assumptions must apply.
(c) Each tray is assumed to represent an ideal equilibrium stage.
(d) Hold-up effects in the column are assumed to be negligible.

The program has been developed for operation in demand mode from a computer terminal. It may also be run in batch mode, by means of a card deck, if necessary.

The program consists of a mainline routine, BATCH, and the following subroutines:

Subroutine	Purpose
TEACH	Provides a brief description of the program.
XDSTEP	Steps off stages on the x,y diagram.
BINBPT	Computes bubble points for equilibrium data calculations, using Antoine's equation and a Newton-Raphson convergence procedure.
XYALFA	Supplies relative volatilities corresponding to vapour or liquid mixture compositions. Values are extracted from an internally compiled array using linear interpolation.
SIMPS	Simpson's Rule numerical integration.
SEARCH	Part of a standard iterative search procedure.

The mainline routine reads in data, performs all calculations and prints out results.

For operation in either batch or demand mode, the program must be mapped as follows:

```
@MAP DISTFILE.BATCHM,.BATCH
```

where element BATCHM contains the following statements

```
IN DISTFILE.BATCH
LIB DISTFILE.
```

Note: Segmented mapping does not reduce the core requirements of this program appreciably.

7.12 Operation of Program

7.121 Graphical representation of results: A plotting routine is available for constructing a McCabe-Thiele x-y diagram representing a computer solution to a problem. The program, BATCH, automatically writes the necessary data to a file, which is added to the plotter runstream through the ERTRAN facility. The file-name used by the plotting routine is BDPLOT. This file must be assigned to the run, with the

identifying WRITE-key 11, before execution of BATCH.

The following two instructions will correctly assign the data file BDPLOT, and must be entered at the start of the runstream if the plotting facility is to be used.

@ASG,UP BDPLOT.
@USE 11,BDPLOT.

7.122 Execution of BATCH in demand mode, with conversational interaction: The following is a description of the full conversational operating procedure for BATCH.

Note: U: - User instruction
P: - Program response

Where options are available to the user, reference is made to a subsequent section where the options are explained.

To start the program:

U: @XQT DISTFILE.BATCH

P: ANSWER ALL QUESTIONS YES OR NO.
ALL INPUT IN FREE FORMAT.
DO YOU KNOW HOW TO USE THIS PROGRAM?

U: YES or NO

P: THIS IS A BINARY BATCH DISTILLATION SIMULATION PROGRAM.
SYSTEMS OPERATING UNDER EITHER VARIABLE OR CONSTANT REFLUX CONDITIONS CAN BE CONSIDERED.
THE PROGRAM IS LIMITED TO SYSTEMS FOR WHICH NORMAL MCCABE-THIELE ASSUMPTIONS HOLD.
PLOTTING ROUTINES ARE AVAILABLE FOR CONSTRUCTING MCCABE-THIELE X-Y DIAGRAMS FOR EITHER OPERATION MODE.

REFER TO THE DISTILLATION PACKAGE MANUAL, SECTION 1, FOR FURTHER INFORMATION.

The program then continues with:

P: MODES OF OPERATION: (1) VARIABLE REFLUX
(2) CONSTANT REFLUX
ENTER 1 OR 2 FOR REQUIRED PROCEDURE.

U: 1 or 2 (See section 7.131)(integer)

P: DO YOU WANT TO ADD A DATA ELEMENT?

U: YES - conversational facility is suppressed and a previously prepared data element is added to the run stream. (See section 7.123.)

or NO

P: ENTER NAMES OF COMPONENTS ... 1 PER LINE,MAX 30 CHARACTERS EACH.

U: COMPONENT NAME1 (light component)
COMPONENT NAME2 (heavy component) (See section 7.132.)

P: ACCEPTED PRESSURE UNITS: 1=MMHG,2=PSIG,3=PSIA,4=KPA,5=ATM. ENTER CONDENSER PRESSURE AND UNIT CODE. (EG. 760.0,1)

U: Pressure, Unit code (See section 7.133) (real, integer)

P: WILL ANTOINE CONSTANTS OR EQUILIBRIUM DATA BE SUPPLIED? ENTER ANT OR EQU.

U: ANT or EQU (See section 7.134 for equilibrium-data specification options.)

P: ENTER NO OF THEORETICAL PLATES IN COLUMN. (MAX=26)

U: Number of theoretical trays. (See section 7.135)(integer)

P: ENTER COMPOSITION AND QUANTITY OF INITIAL CHARGE.

U: Composition, Quantity. (See section 7.136)(real,real)

P: WILL (1) BATCH CYCLE TIME,OR (2) VAPOUR BOIL-UP RATE BE SPECIFIED. ENTER 1 OR 2.

U: 1 or 2 (See section 7.137)

The remaining data-requests depend on whether varying reflux or constant reflux operation is considered. (See section 7.131.)

Finally, for either mode of operation:

P: IS A (1) TOTAL OR (2) PARTIAL CONDENSER USED?
ENTER 1 OR 2.

U: 1 or 2 (See section 7.135) (integer)

The program proceeds with the calculations and if no errors are detected the next output is the result table. (See section 7.14.)

7.123 Execution of BATCH in demand mode, with data element addition: The user indicates that a data element is to be used. The runstream proceeds as follows:

P: ADD YOUR DATA ELEMENT.

U: @ADD DISTFILE.DATA - where DATA contains all the data required for the run, set out in free format. A typical data element is shown in Table 7.1 with each entry briefly described. The data indicate the options required. Further output, excluding error messages, is suppressed until the problem is solved. A table of results is then printed out. (See section 7.14.)

7.13 Selection of optional data inputs

7.131 Variable or constant reflux operation:

(a) U: 1 (Variable reflux)

This number is used by the program to identify the correct calculation procedure and final data requests. For variable reflux these are:

P: ENTER DISTILLATE COMPOSITION.

U: Required product composition as mole fraction of light component. (real)

P: ENTER FINAL RESIDUE CONCENTRATION.

U: Final residue concentration as mole fraction of light component. (real)

For variable reflux operation both the above compositions must be specified. If either is unknown an estimate should be made that can be adjusted later on the basis of computed batch cycle time or boil-up rate.

TABLE 7.1

DATA ELEMENTS FOR OPERATION OF BATCH

AS DESCRIBED IN SECTION 7.123

VARIABLE REFLUX OPERATION

PROBLEM-DATA ELEMENT

1	CARBON DISULPHIDE	Component Names
2	CARBON TETRACHLORIDE	
3	1.,5	Condenser Pressure, unit code
4	EQU	Equilibrium data to be specified
5	NO	Constant relative volatility not assumed
6	19	No. of data pairs
7	@ADD DISTFILE.BATEQU	Data element (see below)
8	6	No. of theoretical trays
9	0.4,50.	Feed composition, quantity (kg moles)
10	2	Vapour boil-up rate specified
11	10.	Vapour boil-up rate (kg moles/hr)
12	0.95	Distillate product composition
13	0.08	Final residue concentration
14	1	Total condenser

EQUILIBRIUM-DATA ELEMENT

	x, y
1	0.05,0.14
2	0.10,0.25
3	0.15,0.34
4	0.20,0.425
5	0.25,0.49
6	0.30,0.55
7	0.35,0.595
8	0.40,0.645
9	0.45,0.69
10	0.50,0.725
11	0.55,0.755
12	0.60,0.79
13	0.65,0.82
14	0,70,0.85
15	0,75,0.877
16	0.80,0.90
17	0,85,0.925
18	0,90,0.95
19	0,95,0.975

(b) U: 2 (Constant reflux)

The program responds as follows:

P: DO YOU WANT TO SPECIFY AN AVERAGE DISTILLATE COMPOSITION?

Two user-responses are allowed

(i) U: YES

P: ENTER AVERAGE VALUE.

U: Required product composition as mole fraction of light component. (real)

(ii) U: NO

P: ENTER FINAL RESIDUE CONCENTRATION.

U: Final residue concentration as mole fraction of light component. (real)

The average distillate composition and the final residue concentration are interdependent. Either may be specified with the unknown value determined by the program.

Finally the constant reflux ratio must be specified.

P: ENTER REFLUX RATIO.

U: Reflux ratio. (real)

7.132 Component names: The length of the component names is limited by the size of the array TITLES(2,5). The names must be entered on separate lines for reading in 5A6 format.

7.133 Condenser Pressure: The specified condenser pressure is assumed to be constant through the column. Pressure drop across trays is not allowed for because the effect is generally small and is not significant in graphical solution procedures. The value is used only if equilibrium data, bubble points, and dew points are computed using Antoine's equation. (Section 7.134.)

The pressure is entered as a real variable and the unit codes as an integer, both in free format.

7.134 Equilibrium data: Three options are available:

(a) U: ANT

P: ENTER ANTOINE CONSTANTS.

U: A,B,C - for light component. (all real)
A,B,C - for heavy component. (all real)

P: ENTER BUBBLE PT OF LIGHT COMPONENT.

U: Bubble point of light component in ^{o}C at specified condenser pressure. (real)

The program uses these data to compute vapour-liquid equilibrium data and corresponding relative volatilities, using Antoine's equation and Raoult's law. The specified bubble point is used as a starting value by the Newton-Raphson convergence procedure in subroutine BINBPT.

(b) U: EQU

P: IS THE RELATIVE VOLATILITY A CONSTANT?

Two replies acceptable:

(i) U: YES.

P: ENTER CONSTANT RELATIVE VOLATILITY VALUE

U: Constant relative volatility. (real)

Program computes vapour-liquid equilibrium data using this value.

(ii) U: NO

P: ENTER NO OF SETS OF DATA TO BE SUPPLIED.
DO NOT INCLUDE X=0.,Y=0. OR X=1.,Y=1.

U: Number of x,y data pairs to be supplied, preferably twenty or more for good accuracy. (integer)

P: ADD EQUILIBRIUM-DATA ELEMENT.

U: @ADD DISTFILE.EQUDAT

where element EQUDAT contains the x,y data pairs, one per line in free format in order of increasing concentration. (See example element, Table 7.1.)

7.135 Number of theoretical trays in column: The number is entered as an integer and must not include either the reboiler or the condenser. The program adjusts the number to include a partial reboiler, and a partial condenser if one is specified, by adding one theoretical stage per device to the specified number of trays.

7.136 Composition and quantity of initial charge: Values are entered as the mole fraction of the light component and the total feed in kg.moles.

7.137 Batch cycle time or vapour boil-up rate: These values are interdependent. The program calculates the variable not specified and adjusts the result table accordingly. The computer response to a user input of either 1 or 2 is

P: ENTER BATCHTIME OR BOIL-UP RATE.

U: Known variable entered. (real)

7.14 Result Table

On completion of the calculations a table of results is printed out. The format depends on the options specified originally by the user, and all results are printed out on the terminal in demand mode. The construction of the complete x,y diagram on the CALCOMP plotter can be initiated from the terminal as shown in section 7.54.

7.15 Execution of BATCH in batch mode.

The same runstream used in demand mode is required for batch mode execution. The run can be entered from the

terminal or by means of a card deck. The following instructions are required in addition to the normal run identification card:

```
@ASG,AX DISTFILE.
@XQT DISTFILE.BATCH
@ADD DISTFILE.BATDATA
```

Element BATDATA must contain the instructions selecting the required procedure, and the problem data as shown in Table 7.1. The conversational output should be suppressed. (See p.195.)

7.16 Error Messages

Origin	Error Message	Description
BATCH	RESIDUE COMPOSITION = ... DISTILLATE COMPOSITION SET TOO LOW.	If $X_{D_{AVE}}$ is specified for a constant reflux operation, the final residue composition may fall below 0,001, which is the minimum allowed.
	SPECIFICATION ERROR ... RESTART PROGRAM	Error in data input format or limits are exceeded, e.g. mole fraction > 1,0 P_T > 10 atm. Illegal reply to a program question.
	ARRAY DIMENSION SIZE EXCEEDED	Equilibrium data parameter NPEQ, or integration parameter NPBS too small.

7.17 Dimension parameters and specification limitations

(a) Dimension parameters.

There are three parameters associated with BATCH.

(1) NPEQ - equilibrium data arrays.

Internal factors DELXAL and NDAT, that generate equilibrium x,y data pairs, set a minimum value of NPEQ = 51.

In general: NPEQ ≥ NDATA+2 where NDATA is the number of data pairs supplied, excluding end points 0.,0. and 1.,1.

(2) NPTC - number of trays.

The coordinates of points defining the stepped stages on a McCabe-Thiele x,y diagram are stored for plotting purposes. The arrays X and Y are dimension NPTC where

$$NPTC \geq 2(NT+2)+1$$

where NT is the specified number of trays excluding the reboiler and partial condenser. If a total condenser is specified

$$NPTC \geq 2(NT+1)+1$$

Present value of 55 restricts BATCH to systems with 25 trays and a partial condenser or 26 trays with a total condenser.

(3) NPBS - numerical integration data arrays.

Internal counter NINT determines the number of increments over which integration is performed. In general:

$$NPBT \geq NINT+1$$

where NINT is defined in a DATA statement in BATCH. Other arrays have fixed dimensions to accommodate two components.

(b) Specification limitations.

(1) System pressure

In all the binary system programs in this package, the specified system pressure is assumed to be constant throughout the column for purposes of data calculations. Antoine's equation and other correlations included in the programs (see sections 7.2 and 7.3) are in general limited to system pressures of less than 10^3 kPa. Hence the following limitation is imposed.

Specified total pressure: $P_T \leq 10^3$ kPa (10 atm)

(2) Number of trays in column.
See (a)(2) above.

7.2 BINARY CONTINUOUS DISTILLATION IN TRAY COLUMNS

7.21 Application

The program BINARY designs binary continuous distillation systems using either the McCabe-Thiele, or the Ponchon-Savarit, procedure. The dimensions of the column are not calculated hence superficial liquid and vapour flowrates are not considered. The Ponchon-Savarit procedure will consider simple columns whereas the McCabe-Thiele procedure is dimensioned to accommodate complex columns with up to five feed and five side streams. Murphree tray efficiencies may be specified for either procedure, and either total or partial condensers may be considered. Open steam may replace a partial reboiler in the McCabe-Thiele procedure.

Both design procedures compute the number of trays required to achieve a desired separation at a specific reflux ratio. The feed and side streams are positioned to yield the minimum number of trays possible for given operating conditions.

The mainline program BINARY has the following subroutines associated with it.

Subroutine	Purpose
TEACH	Provides a brief description of the program.
READBD	Reads in all data conversationally or from a data element. Calls CONDEN to determine distillate product compositions for either total or partial condenser operation. Calls FLASHB to determine feed condition for Ponchon-Savarit designs.

CONDEN	Distillate product composition calculations.
FLASHB	Flash calculation for Ponchon-Savarit designs. Calls EQUIL.
EQUIL	Determines equilibrium compositions at flash temperature of a two phase feed stream.
MULTFS	Determines operating conditions for complex columns in McCabe-Thiele designs.
SIMPLE	Same as MULTFS but for simple columns.
PONSAV	Constructs enthalpy-concentration diagram and x,y diagram operating lines for Ponchon-Savarit designs. Calls HTCAP, HFIND, RMIN, OPLINE and FLOWS.
HTCAP	Vapour and liquid stream enthalpies calculated. Calls HTSOLN.
HTSOLN	Determines heat of solution value, if data are supplied for use in liquid stream enthalpy calculations.
HFIND	Linear interpolation search for enthalpy of either vapour or liquid stream of given composition from data compiled in HTCAP.
RMIN	Computes minimum reflux ratio for Ponchon-Savarit design.
OPLINE	Defines operating lines in Ponchon-Savarit x,y diagram.
FLOWS	Computes flowrates and temperatures through Ponchon-Savarit columns. Calls AVEMW.
AVEMW	Computes average molecular weight of streams for mass flowrate calculations. Mass flowrates required in design of packed columns. (See section 7.3.)
CHEC	Searches for pinch points on x,y diagram before attempt is made to step off equilibrium stages.

STEPD	Steps off equilibrium stages on x,y diagram for either design procedures. Calls PSOPXY and FFIND.
PSOPXY	Performs material balance calculation for STEPD in stepping procedure for Ponchon-Savarit designs.
FFIND	Determines flowrates of streams leaving each tray in a Ponchon-Savarit column from data compiled in FLOWS. Determines temperature on each stage as well. Linear interpolation used.
BINRES	Prints result table for BINARY.

BINBPT,BINDPT,XYALFA and SEARCH are also used by BINARY and its subroutines. These are discussed in section 7.11.

In order to minimise core requirements, BINARY must be mapped as follows:

```
@MAP DISTIFILE.BINARYM,.BINARY
```

where element BINARYM contains the following statements:

```
1           SEG BINARY1
2           IN BINARY
3           SEG READBD1*,(BINARY1)
4           IN READBD
5           SEG SIMPLE1*,(BINARY1)
6           IN SIMPLE
7           SEG PONSAV1*,(BINARY1)
8           IN PONSAV
9           SEG RMIN1*,(PONSAV1)
10          IN RMIN
11          SEG OPLINE1*,(PONSAV1)
12          IN OPLINE
13          SEG FLOWS1*,(PONSAV1)
14          IN FLOWS
15          SEG CHEC1*,(BINARY1)
16          IN CHEC
17          SEG STEPD1*,(BINARY1)
18          IN STEPD
19          SEG PSOPXY1*,(STEPD1)
20          IN PSOPXY
21          SEG FFIND1*,(STEPD1)
22          IN FFIND
23          SEG BINRES1*,(BINARY1)
24          IN BINRES
25          LIB DISTFILE.
```

7.22 Operating the program

7.221 Graphical representation of results: A plotting routine is available to produce a standard McCabe-Thiele or Ponchon-Savarit diagram representing the computer solution to a problem. Key points on each diagram are marked to simplify interpretation of the results. To make use of this facility the data file BDPLOT must be assigned to the run as described in section 1.21, before execution of BINARY. The WRITE key in BINARY is 11, hence the necessary instructions are

```
@ASG,UP BDPLOT.
@USE 11,BDPLOT.
```

The WRITE key is defined in a DATA statement in BINARY and may be changed if necessary.

7.222 Execution of BINARY in demand mode, with conversational program user interaction: Execution started by:

```
U: @XQT DISTFILE.BINARY
```

The program responds as shown in section 7.122. If the user replies NO to the first question, the following description of the program is printed out.

```
P: THIS IS A BINARY CONTINUOUS DISTILLATION DESIGN PROGRAM.
   IT WILL COMPUTE THE NUMBER OF THEORETICAL STAGES RE-
   QUIRED TO ACHIEVE A SPECIFIED TOP PRODUCT PURITY.
   A MURPHREE PLATE EFFICIENCY FACTOR MAY BE SPECIFIED.
   THE DIMENSIONS OF THE COLUMN REQUIRED ARE NOT CALCULATED.
   SYSTEMS FOR WHICH EITHER MCCABE-THIELE OR PONCHON-
   SAVARIT ASSUMPTIONS HOLD,MAY BE CONSIDERED.
   PLOTTING ROUTINES ARE AVAILABLE FOR CONSTRUCTING STAN-
   DARD DIAGRAMS FOR EITHER DESIGN TECHNIQUE.

   REFER TO THE DISTILLATION PACKAGE MANUAL,SECTION 2,
   FOR FURTHER INFORMATION.
```

The execution is then terminated as it is unlikely that a successful run would be completed without preparation. The normal runstream proceeds as follows if the first question is answered YES.

P: DESIGN TECHNIQUES: (1) MCCABE-THIELE
(2) PONCHON-SAVARIT
ENTER 1 OR 2 FOR REQUIRED PROCEDURE.

U: 1 or 2 as required. This is read in as counter NOP which selects the appropriate options.

The next program response concerns data element addition as shown in section 7.122. The conversational output can be suppressed and a data element added. A sample data element is shown in Table 7.21.

The conversational runstream continues as shown in section 7.122 up to and including the equilibrium data specification. In BINARY, however, equilibrium temperature data corresponding to the composition data, must be supplied if option EQU is used. This is described in section 7.231.

TABLE 7.21

PROBLEM-DATA ELEMENT FOR MCCABE-THIELE DESIGN

1	METHANOL	Component names
2	WATER	
3	760.0,1	Column pressure and unit code.
4	ANT	Antoine constants specified.
5	7.87863,1473.11,230.	Antoine constants
6	7.96681,1668.21,228.	
7	64.7,100.	Pure component boiling points.
8	0.95,38.1	XD,DTOT
9	1.16	R
10	PAR	Partial condenser specified.
11	0.4,0.6	Vap & Liq Fraction in 2 phase distillate
12	NO	'IS XD OVERALL COMPOSITION?'
13	LIQ	Phase defined by XD
14	1	No. of feed streams
15	0	No. of side streams
16	0.39,98.1	Z,FTOT
17	1.04	Q
18	1	Open steam replaces reboiler
19	0.8	Efficiency factor (Murphree tray efficiency)

After supplying equilibrium data the runstream proceeds as follows:

P: ENTER DISTILLATE COMPOSITION AND WITHDRAWAL RATE.

U: XD,FTOT (mole fraction, hg moles/hr, both real)

P: ENTER REFLUX RATIO.

U: R (real)

P: TOTAL OR PARTIAL CONDENSER?...ENTER TOT OR PAR.

U: TOT or PAR. (See section 7.232.)

Counter NOP selects the options corresponding to the design procedure chosen. For NOP=1 the runstream proceeds at (a) below; for NOP=2, the runstream continues at (b).

(a) NOP=1 (McCabe-Thiele design only.)

P: ENTER NO.OF FEED STREAMS.

U: NF (integer)

P: ENTER NO. OF SIDE STREAMS.

U: NS (integer)

If NF+NS=1 (simple columns) the runstream continues at (b) below.

For NF+NS>1 (complex columns):

P: ENTER FEED STREAM COMPOSITIONS.

U: Z_{F_k} k=1,NF mole fraction of light component in each feed. (ZF)

P: ENTER FEED Q VALUES.

U: Q_{F_k} k=1,NF feed q values in same order as Z_F (QF)

P: ENTER FEED RATES

U: F_{F_k} k=1,NF feed flowrates kg mole/hr. (FTOTS)

P: ENTER SIDE STREAM COMPOSITIONS

U: Z_{S_k} k=1,NS (ZS)

P: ENTER SIDE STREAM Q.VALUES.

U: Q_{S_k} k=1,NS (QS)

P: ENTER SIDE STREAM WITHDRAWAL RATES.

U: F_{S_k} k=1,NS (STOTS)

The positions of the feed and side streams in the column are determined by the program to minimise the total number of trays required. Hence the above data may be inserted in any order.

(b) NOP=2 (Ponchon-Savarit designs) or McCabe-Thiele simple columns:

P: ENTER FEED COMPOSITION AND FEED RATE

U: Z,FTOT

A feed q-value (McCabe-Thiele) or a feed temperature (Ponchon-Savarit) must be specified as follows, (See section 7.233):

P: ENTER FEED Q VALUE.

U: Q

or P: ENTER TEMPERATURE OF FEED..(DEG C).

U: TF

*P: IF THE HEAVY COMPONENT IS WATER AND OPEN STEAM OPERATION IS REQUIRED, ENTER 1...IF NOT ENTER 0

U: 0 (partial reboiler)

or 1 (open steam).

P: ENTER MURPHREE STAGE EFFICIENCY FACTOR (=1.0 IF PERFECT TRAYS ARE ASSUMED)

U: E_{V_m} (real)

This is the last entry required for McCabe-Thiele designs.

* McCabe-Thiele designs only.

Enthalpy data are required for Ponchon-Savarit designs only and must be added to the runstream in a data element.

P: ADD ENTHALPY DATA ELEMENT.

U: @ADD DISTFILE.ENTHDAT (See section 7.234.)

P: Feed q-value and temperature data are printed out for inspection.

This is the final entry required for Ponchon-Savarit designs. If no errors are detected during computation, the next output is the result table (section 7.24).

7.23 Input of problem specifications.

7.231 Equilibrium data: The procedure and options are described in section 7.134 with the following differences:

(1) Temperature data corresponding to the equilibrium data, are required for Ponchon-Savarit designs. Zeros or fictitious values must be included if a McCabe-Thiele design is required. The data are read in as x_i, y_i, T_i, with one set of three values per line. The table starts at low concentrations with the values referring to the light component.

(2) The constant relative volatility option is not available in Ponchon-Savarit designs because temperature data are required.

(3) If Antoine constants are specified, the bubble points of both components must be specified; light component value first. Temperature data are then generated using the bubble point routine.

7.232 Total or Partial condenser operation: Specification of a total condenser requires no further input related to the distillate product. If, however, a partial condenser is used, and a two phase product withdrawn, the previously entered composition value XD must be defined as one of three options:

(a) Overall distillate product composition

(b) Vapour phase composition

(c) Liquid phase composition

After specifying a partial condenser, the runstream proceeds as follows:

P: ENTER FRACTION OF DISTILLATE WITHDRAWN AS VAP AND LIQ RESP.
FOR ALL VAPOUR DISTILLATE ENTER 1.,0.

U: Fraction of each phase, vapour phase value first.
e.g. 0.5,0.5 for equimolar withdrawal of each phase.

If the distillate is to be withdrawn as a vapour only the runstream continues with:

*P: ENTER NO. OF FEED STREAM (McCabe-Thiele designs)

or P: ENTER FEED COMPOSITION AND RATE (Ponchon-Davarit design)

For a two-phase distillate product system the program responds with

P: IS XD AN OVERALL COMPOSITION VALUE

U: YES (runstream proceeds as shown above*)

or U: NO

P: IS XD THE COMPOSITION OF THE VAPOUR OR LIQUID FRACTION ENTER VAP OR LIQ

U: VAP

or LIQ (run proceeds as shown above*)

A two-phase distillate product is allowed in McCabe-Thiele designs only.

7.233 Feed temperature or q-value specifications: It is customary to specify the feed condition in terms of a q-value in McCabe-Thiele tutorial problems. The program BINARY requires a specified q-value for each feed and side stream in a McCabe-Thiele design.

In Ponchon-Savarit designs the knowledge of enthalpy data permits the determination of the feed condition by flash calculation. To perform a flash calculation, the overall composition and temperature of the feed must be known. The feed q-value is determined from equilibrium data if the temperature of the feed is found to lie between the dew point and the bubble point. The following expression is used.

$$q = \frac{y - Z_F}{y - x} \tag{7.21}$$

where y and x are the equilibrium concentrations at the feed temperature.

If the feed temperature indicates that the feed is either superheated or supercooled at the column pressure, the calculation of q is performed according to its definition as the heat required to convert one mole of feed from its condition H_F to a saturated vapour, divided by the molal latent heat

$$q = \frac{H_V - H_F}{H_V - H_L} \tag{7.22}$$

where H_V and H_L are the enthalpies of the feed at its dew point and bubble point respectively.

The above two equations may be used to check computer results.

7.234 Enthalpy calculations: In Ponchon-Savarit designs the effect of enthalpy on the vapour and liquid flow profiles is considered. The program constructs an enthalpy-concentration diagram for the system, and uses the standard graphical construction procedure to determine the positions of the operating lines on the x-y diagram.

The vapour and liquid enthalpy curves are determined using the heat capacity correlation of Kobe [55] and Watson's [56] correlation for extrapolating latent heat data with

respect to temperature. Heat of solution effects may be included if data are available.

(a) Kobe heat capacity polynomial.

$$Cp_i = a_i + b_i T_j + c_i T_j^2 + d_i T_j^3 \qquad (7.23)$$

Coefficients a,b,c and d for a number of compounds are listed in Hougen, Watson and Ragatz, Part II [57]. Coefficients for other compounds can be found in the original articles of Kobe [55], or may be determined by applying a least squares fit of the above polynomial to available data. The number of compounds listed in Hougen et al. should be sufficient for setting tutorial problems dealing with a wide range of systems. Coefficients b_i,c_i and d_i are quoted as (coeff. x 10^{n_k}). The user enters the coefficients and the corresponding negative exponents ($-n_k$) separately (section 7.234(f)). The program then adjusts the decimal point accordingly. This procedure simplifies the use of free-format for data inputs.

(b) Liquid enthalpy

Enthalpy data are computed in subroutine HTCAP using the definite integral of the above polynomial.

$$h_i = \int_{T_b}^{T_B} Cp_i dT \qquad (7.24)$$

where T_b = a specified base temperature.

Coefficients taken from Hougen et al. are derived for temperature in K and heat capacity in cal $gmole^{-1}$. A number of conversion factors are included to enable data to be specified in convenient units. These are discussed in section 7.235.

(c) Heats of solution

If heat of solution data are available, these may be included in the liquid-enthalpy calculations. Heat of solution data for a few systems are available in International Critical Tables, Vol.5,p.159. These data are quoted as composition vs. heat of solution, determined at a given temperature. The temperature at which these data are available must be specified as the base temperature, T_b, for enthalpy calculations.

Subroutine HTSOLN determines the heat of solution of a mixture of composition x using linear interpolation between points in the given data table. Conversion factors adjust the units to correspond with those used in HTCAP. (See section 7.235 below.)

Note: Enthalpy correlation charts often include heat of solution effects. If coefficients for Equation (7.23) are derived from such charts heat of solution data should not be provided separately.

(d) Vapour Enthalpy

Enthalpy data are determined for the vapour phase by adding the latent heat, calculated at the vapour temperature, to the enthalpy of a liquid of the same composition, calculated at the vapour temperature [57]. If the vapour is superheated, the latent heat is calculated at the dewpoint of the vapour.

One value of the latent heat of vaporization for each component must be specified at the normal boiling point of the pure component. Critical temperatures and normal boiling points of each component must be specified in ^{o}C.

Watson's correlation [56] is used to extrapolate a single latent heat value with respect to temperature. The correlation may be used for temperatures within 10 K of T_c [58]:

$$LH_{i_k} = LH_{i_{BP}} \left[\frac{T_{C_i} - T_k}{T_{C_i} - T_{BP}} \right]^{0,38} \qquad (7.25)$$

(e) Mixture Enthalpy

The enthalpy of liquid mixtures is computed as

$$h_L = \Sigma x_i h_i + H_S \qquad (7.26)$$

The enthalpy of vapour mixtures is computed as

$$H_V = \Sigma y_i H_i$$

where $$H_i = h_i + LH_{i,T_D} \qquad (7.27)$$

with h_i computed at the temperature of the vapour T_D.

(f) Data required:

A data element containing all the required enthalpy data must be set up before the program is run. Data are inserted in free format, subscript 1 indicates the light component. Table 7.22 shows problem and enthalpy of data elements for Ponchon-Savarit designs.

Data element ENTHDAT

a_1, b_1, c_1, d_1	Values entered as a, bx10^2, cx10^5, and
a_2, b_2, c_2, d_2	dx10^9. See Hough, Watson & Ragatz [57].
NBCP, NCCP, NDCP	$-n_k$, k=b,c,d; exponents corresponding to b,c and d respectively. E.g. -2,-5,-9
CPCONV, LHCONV,	See section 7.235 below.
HSCONV, HEVABS	" " " "
TBASE, TCONV	" " " "
LH_1, TBP_1, TC_1	Latent heat, normal boiling point and
LH_2, TBP_2, TC_2	critical temperature for each comp.
$MW_1, CPMW_1, LHMW_1, HSMW_1$	See section 7.235 below
$MW_2, CPMW_2, LHMW_2, HSMW_2$	" " " "

NHS — Number of heat of solution data pairs.

XHS_k, $HSOLN_k$ k=1,NHS — Composition - heat of solution data if available.

Note: If heat of solution data is not used NHS must be entered as zero.

TABLE 7.22

PROBLEM DATA ELEMENT FOR PONCHON-SAVARIT DESIGNS

1	METHANOL	Component names
2	WATER	
3	760.0,1	Column pressure and code
4	ANT	Antoine constants specified
5	7.87863,1472.11,230	Antoine constants
6	7.96681,1668.21,228	
7	64.7,100.	Pure component boiling points
8	0.915,38.1	XD,DTOT
9	1.16	R
10	TOT	Total condenser specified
11	0.36,98.1	ZF,FTOT
12	57.8	Feed temperature
13	1.0	Efficiency factor (for ideal trays)
14	@ADD DISTFILE.ENTHDAT	Enthalpy data element.

ENTHALPY DATA ELEMENT ENTHDAT

1	4.55,2.186,-0.291,-1.92	Kobe function coefficients
2	7.7,0.04594,0.2521,-0.8587	
3	-2,-5,-9	Exponents relating to coefficients
4	1.8,1.8,430.,-1.	CPCONV,LHCONV,HSCONV,HEVABS
5	19.69,273.	Base temperature, TCONV
6	262.79,64.7,240.	LH_i,T_{BPi},T_{Ci}
7	540.,100.,374.15	
8	32.04,1.0,32.04,1.0	MW_i $CPMW_i$ $LHMW_i$ $HSMW_i$
9	18.02,1.0,18.02,1.0	
10	0	NHS

7.235 Conversion factors: Enthalpy data are usually quoted in a variety of units. For example, coefficients for the heat capacity polynomial from Hougen et al. requires

temperature data in K and computes C_p data in cal $gmole^{-1}$, whereas many of the coefficients shown in the original work of Kobe require temperatures to be specified in ^{o}C.

The calculations performed in the subroutines HTCAP and HTSOLN are independent of the calling routine. Conversion factor ENCONV in PONSAV converts all data returned by HTCAP to the S.I. system of KJ $Kgmole^{-1}$. The value of ENCONV presently inserted in PONSAV programs assumes that enthalpy data are returned in Btu/lb mole. CONFAC converts heat duty units from kJ/hr to kW.

The conversion factors used in HTCAP and HTSOLN must therefore convert the units in which data is supplied to the F.P.S. system. This double conversion has been adopted because many example problems in the literature use the British system of units. By setting CONFAC and ENCONV = 1 in the mainline, results are printed in the British system and this enables the computer solution to be compared to the text solution with no further calculations.

The conversion factors are supplied in three sets

(a)	CPCONV,CPMW	Heat capacity correlation
(b)	LHCONV,LHMW	Latent heat correlation
(c)	HSCONV,HSMW	Heat of solution data.

Each set is used in the same way, as follows:

Consider latent heat data found in the literature in cal gm^{-1} [58]. To convert values to Btu/lb mole set LHCONV =1.8 and LHMW = molecular weight of component. These factors will then convert cal gm^{-1} to Btu lb^{-1} and weight to moles respectively. If either unit is correct the corresponding conversion factor is set equal to 1,0.

Other factors shown in the data list are defined as follows:

TBASE — Base temperature for enthalpy calculations. If heat of solution data are supplied, the temperature at which this is quoted is used as the base temperature. Units: ^{o}C.

TCONV — Used to convert temperatures in ^{o}C to K for use in the C_p correlation. If coefficients are specified for a polynomial in ^{o}C, TCONV is set equal to zero.

HEVABS — Specified as either -1 if heat of solution is exothermic or +1 if heat of solution is endothermic.

MW_1, MW_2 — Molecular weight of each component required in mainline program PACKED.

7.24 Results

On completion of the calculations a table of results is printed on the terminal in demand mode, or on the line printer in batch mode. The format of the table is determined by the type of problem solved (McCabe-Thiele or Ponchon-Savarit). The main section of the table contains the standard results required in most tutorial problems. Additional tables of equilibrium and operating line data for manual construction of an x,y diagram is printed on request as follows:

P: DO YOU WANT OPERATING DIAGRAM DATA TABLES?

U: YES or NO

This reply must be included in batch mode runstreams as well.

A McCabe-Thiele x,y diagram or a complete Ponchon-Savarit operating diagram may be constructed on the CALCOMP plotter using the instructions shown in section 7.55.

7.25 Execution of BINARY in batch mode

The following statements are required to execute BINARY in batch mode:

```
@ASG,AZX DISTFILE
@XQT DISTFILE.BINARY
@ADD DISTFILE.BINDATA
```

where BINDATA is a data element which must conform to the demand mode runstream. The conversational output should be suppressed in batch mode.

If a plot is required the plotting data file must be assigned before the @XQT statement, as shown in section 7.221. The plotting routine can be started in batch mode as shown in section 7.63, but this should not be done until the results have been examined.

7.26 Error messages

Certain subroutines are called both by BINARY and by PACKED, and hence the error messages associated with these subroutines are common to both programs.

Origin	Error Message	Description
BINARY	ILLEGAL SPECIFICATION.ERROR TERMINATION	Reply to question other than YES or NO.
SIMPLE, PONSAV	ERROR...BTOT OR XB IS LESS THAN 0.0 OR XB IS GREATER THAN Z	Flowrate or composition specification causes failure of mass balance
MULTFS	ERROR...TWO STREAMS HAVE SAME COMPOSITION	Two streams, feed or side in complex columns have been specified with equal compositions.
	ERROR... BTOT OR XB IS LESS THAN 0.0	Similar to SIMPLE above.

Origin	Error Message	Description
PONSAV	SPECIFIED REFLUX RATIO IS LESS THAN THE CALCULATED MINIMUM RMIN = RSPEC = A VALUE OF 1.5*RMIN WILL BE USED	(Non terminating message.
CHEC	PINCH POINT AT SIDE (OR FEED) STREAM COMPOSITION	Point of intersection of two operating lines detected above equilibrium curve. Change flowrate or q-value of indicated stream or increase the reflux ratio.
READBD	ERROR IN DATA INPUT...ERROR TERMINATION	Data does not conform to specified format or limitations e.g. $P_T > 10$ atm or a reply to a question is illegal.
	ARRAY DIMENSION SIZE EXCEEDED	Number of data points exceeds dimension parameter.
STEPD	NUMBER OF TRAYS EXCEEDS: PINCH ZONE IS SUSPECTED...	Dimension parameter NPTC is exceeded.
RMIN	ERROR IN RMIN.FEED ENTER ON TOP TRAY	Search for RMIN has determined that Δ_{D_M} lies on or below the vapour enthalpy curve on the enthalpy-conc. diagram.
	ERROR IN RMIN ROUTINE THIE LINE PASSING THROUGH POINT(Z,HF) NOT FOUND	Number of iterations allowed for search is 500. This may be exceeded

	HTEST= HF= Y1= X1= NO.OF ITERATIONS= INCREMENTS= (Y1 and X1 are the composition coordinates of the tie line last tested. ATEST is the iterative enthalpy value that should converge to HF when the tie line passing through HF is found.)	if the convergence limit ACC in RMIN is too small. In this case the displayed increment will probably be very small and HTEST ∿HF.
	ERROR IN RMIN ROUTINE ENTHALPY OF VAPOUR ENTERING PARTIAL CONDENSER NOT FOUND.	Similar to above error. Result of iterative search failure at 500 iterations. ACC is probably too small.
OPLINE	ERROR...NO. OF INTERVALS CALCULATED IN OPLINE EXCEED DIMENSION SPECIFICATIONS NO.OF INTERVALS=	Refers to parameter NPPS.
	ERROR IN OPLINE LOOP 'n' EXCEEDED 500 ITERATIONS.	Similar to search errors in RMIN. Increase size of ACC in OPLINE.
FLOWS	TOTAL NUMBER OF OPERATING LINE DATA POINTS EXCEEDS DIMENSION SPECIFICATIONS	NPPS should be increased.
	NEGATIVE LIQUID (OR VAPOUR) FLOW DETECTED...(self explanatory)	

7.27 Dimension parameters and specification limitations

(a) Dimension parameters.

There are four parameters associated with BINARY:

(1) NPEQ - equilibrium data arrays. (See section 7.17.)

(2) NPTC - number of trays. (See section 7.17.)

(3) NPEN - Enthalpy data arrays.

In computing enthalpy-concentration data for constructing curves on the Ponchon-Savarit diagram, the interval between points sets the size of the array required. Intervals DELX and DELY in PONSAV of 0,05 mole fraction produce smooth curves for most ideal systems. The parameter can thus be defined as

$$NPEN \geq 1.+DELX^{-1} \text{ or } 1.+DELY^{-1}$$

whichever is larger if DELX $\neq$ DELY.

(4) NPPS - Arrays associated with Ponchon-Savarit Calculations.

The COMMON section of block BINCOM is used both in BINARY and in PACKED. Arrays dimensioned NPPS are used in Ponchon-Savarit and packed column calculations, where internal counter NINT determines the number of points defining the top operating line. The size of the interval between points is kept the same for both operating lines to maintain a constant level of accuracy in the integration procedure in PACKED. Hence the number of points defining the bottom operating line could exceed NINT. Also, certain arrays contain data for both operating lines. Thus to ensure that sufficient space is available set NPPS as follows:

$$NPPS \geq 3 \times NINT$$

where NINT is defined in a DATA statement in the mainline routine. Counter NPPSC = NPPS-2 is used to check that the array size is not exceeded during calculations.

Other arrays have fixed dimensions to accommodate two components, and complex columns with five feed and five side streams with associated data.

(b) Specification limitations.

(1) Pressure specification. (See section 7.17.)

(2) Number of trays.

Limitation as described in section 7.17 indicated by error message if the designed number of trays exceeds the dimension parameter NPTC.

(3) Complex columns may contain a maximum of five feed and five side streams. No two streams may have the same composition. The streams can not be positioned in the column in a particular order. The program arranges the streams in order of increasing concentration of the light component, moving up the column.

(4) Either a two phase or a single phase distillate product may be withdrawn from a partial condenser. If a two phase product is withdrawn the user must specify whether the distillate composition refers to a particular phase, or the overall product.

(5) Accuracy limits on convergence procedures in RMIN and OPLINE should not be set too low. The present value (ACC=0.0001 x HF) represents an error of 0,01% of the absolute feed-enthalpy value, in each calculation. This produces satisfactory results for normal ideal systems. Lower values increase the CPU time required and may cause failure of the convergence procedure.

7.3 BINARY CONTINUOUS DISTILLATION IN PACKED COLUMNS

7.31 Application

The program PACKED determines the diameter and height of a bed of packing required to achieve a given separation. The design is based on a Ponchon-Savarit operating diagram and uses Shulman's [61] mass transfer coefficient correlations in the calculation procedure. The design procedure is described by Treybal [9]. Flooding and loading characteristics are determined using the generalised flooding correlation chart of the U.S. Stoneware Co. (see Figure 1, Appendix A).

It is assumed that the column is fitted with a partial reboiler and either a partial or total condenser. A single phase distillate product is withdrawn. The program uses the same procedure for constructing the Ponchon-Savarit diagram as the program BINARY, see section 7.2. Other subroutines associated with PACKED include

Subroutine	Purpose
PACTYP	Determines type of packing required by the user, and calls PAKING to extract the relevant data.
PAKING	Contains packing data for seven packings. (See Appendix A, Tables 1,2 and 3.) The user may add to this data routine if necessary.
FDATA	Fits polynomials to the surves shown on the flooding correlation chart (Figure 1, Appendix A). The user selects a curve by specifying a column pressure drop.
PACRES	Prints out result table on completion of the design.

The physical property package is also required. The physical property subroutines have been designed so that they can be used by other programs developed in the Department of Chemical Engineering, U.C.T. Program information is entered through a calling list and data required by the subroutine are read from a specially prepared data element through the ERTRAN facility. (See section 7.34.)

In order to minimise core requirements, PACKED must be mapped as follows:

```
@MAP DISTFILE.PACKEDM,.PACKED
```

where element PACKEDM contains the following statements:

```
 1          SEG PACKED1
 2          IN PACKED
 3          SEG READBD1*,(PACKED1)
 4          IN READBD
 5          SEG PONSAV1*,(PACKED1)
 6          IN PONSAV
 7          SEG RMIN1*,(PONSAV1)
 8          IN RMIN
 9          SEG OPLINE1*,(PONSAV1)
10          IN OPLINE
11          SEG FLOWS1*,(PONSAV1)
12          IN FLOWS
13          SEG PACTYP1*,(PACKED1)
14          IN PACTYP
15          SEG PAKING1*,(PACKED1)
16          IN PAKING
17          SEG PACRES1*,(PACKED1)
18          IN PACRES
19          LIB DISTFILE.
```

7.32 Operating the program

There are no plotting routines associated with PACKED. The Ponchon-Savarit diagram can be obtained by running the same data in BINARY. This will also yield the number of trays equivalent to the packed height for comparative purposes. These data could be used to calculate the HETP for the system.

PACKED should be executed in demand mode because the user is required to inspect results at intervals during the run. Decisions must be made, based on these results, to direct further computation. These decisions are discussed in the runstream shown below. Subroutine READBD is used by PACKED as well as BINARY to read basic operating data for a problem. The initial part of the runstream is thus similar to that described in section 7.222 for Ponchon-Savarit designs, and requires no further explanation. The brief description of the program provided by TEACH is as follows:

```
P: THIS PROGRAM DESIGNS PACKED COLUMNS FOR DISTILLATION
   OF BINARY SOLUTIONS.
   THE PONCHON-SAVARIT METHOD IS USED TO DETERMINE THE
   OPERATING DIAGRAM.
   THE HEIGHT OF PACKING REQUIRED IN EACH SECTION OF THE
   COLUMN IS DETERMINED BY INTEGRATION,USING SIMPSONS
   RULE.
   MASS TRANSFER COEFFICIENTS ARE COMPUTED USING SHULMANS
   CORRELATIONS AND PACKING DATA.
   DATA FOR CERAMIC RASCHIG RINGS (4 SIZES) AND BERL
   SADDLES (3 SIZES) ARE INCLUDED IN THE PACKAGE.

   REFER TO THE DISTILLATION PACKAGE MANUAL,SECTION 3,
   FOR FURTHER INFORMATION.
```

This terminates the execution as described in section 7.222.

Two data elements may be added to the runstream, with an optional conversational procedure available in each case. The first element concerns the basic problem specifications and is very similar to the element required by BINARY. The second element contains data relating to the type and size of packing required, and other mass transfer calculation details. An example of each element is shown in Table 7.3.

TABLE 7.3

FIRST DATA ELEMENT CONTAINING PROBLEM-DATA

1	METHANOL	Component names
2	WATER	
3	760.0,1	Column pressure and code
4	ANT	Antoine constants specified
5	7.87863,1473.11,230	Antoine constants
6	7.96681,1668.21,228	
7	64.7,100.	Pure component boiling points
8	0.915,38.1	XD,DTOT
9	1.16	R
10	TOT	Total condenser specified
11	0.36,98.1	Z,FTOT
12	57.8	Feed temperature
13	@ADD DISTFILE.ENTHDAT	(See Table 7.22)

SECOND DATA ELEMENT SPECIFYING PACKING AND COMPONENT PROPERY DATA

1	NO	'DO YOU WANT TO USE A DIFFERENT TYPE OF PACKING?'
2	1	Type of packing code (Raschig rings)
3	2	Size of packing code (25,4 mm)
4	NO	'ARE SOLUTION-DENSITY DATA ALSO AVAILABLE?'
5	1,1	Component class (see below)

After the first data element has been read, or these data inserted conversationally, the runstream proceeds as follows:

```
P:FEED Q VALUE AND TEMPERATURE DATA:
  Q:  1.0239        FEED TEMP:  57.80 DEG C
  DEW PT:  91.83 DEG C  BUBBLE PT: 81.88 DEG C
```

(Calculated feed q-value, dew point and bubble point values and the specified feed temperature.)

```
P:DATA ACCEPTED. START OF OPERATING DIAGRAM CALCS.
```

If the specified reflux ratio is too low (see subroutine PONSAV), if the column does not contain a stripping section (see subroutine FLOWS) or if any errors are detected during the construction of the operating diagram, the user is informed here (see section 7.26).

P: OPERATING DIAGRAM COMPLETE.
START OF PACKING CALCULATIONS.

DO YOU WANT TO ADD A SECOND DATA ELEMENT?

U: YES - for a sample data element, see Table 7.3.

or NO - the conversational runstream continues.

P: THE PROGRAM DATA PACKAGE INCLUDES PACKING DATA FOR CERAMIC RASCHIG RINGS AND BERL SADDLES.
DO YOU WANT TO USE A DIFFERENT TYPE OF PACKING?

U: YES: (See section 7.331)

NO:

P: DO YOU WANT TO USE (1) RASCHIG RINGS OR (2) BERL SADDLES?
ENTER 1 OR 2

U: 1 or 2; for 2:

P: THE AVAILABLE SADDLE SIZES ARE:
(1) 12.7,(2) 25.4,(3) 38.1 (MM)
ENTER APPROPRIATE CODE NO.

U: 1, 2 or 3

P: COMPONENT-DENSITY DATA MUST BE AVAILABLE AS DESCRIBED IN THE DISTILLATION PACKAGE MANUAL,SECTION 3.4
ARE SOLUTION-DENSITY DATA ALSO AVAILABLE?

U: YES or NO (See section 7.34)

P: COMPOUND CLASSES ARE:
(1) ALCOHOLS AND WATER
(2) HYDROCARBONS (SAT.,UNSAT. AND AROM.) AND ETHERS
(3) OTHER ORGANICS
(4) INORGANIC COMPOUNDS
ENTER CLASS NO. INTO WHICH THE LIGHT AND HEAVY COMPONENTS FALL
EG. FOR WATER AND NITRIC ACID ENTER 1,4

U: Appropriate reply depending on components in feed. (See section 7.34.)

The runstream now continues conversationally regardless of execution mode. PACKED is designed primarily for demand mode execution where the user is expected to direct the design calculations as shown below.

P: DO YOU WANT TO SPECIFY THE COLUMN DIAMETER?

U: NO: See (b) below (See section 7.332)
or YES: (See section 7.332)

(a) Referring to the flooding correlation chart (see Figure 1, Appendix A), the user may select a diameter on the basis of column pressure drop, as shown below.

P: ENTER COLUMN DIAMETER IN METRES.

U: 1.0

P: OPERATING POINT COORDINATES:
ABSCISSA= .049
ORDINATE= .031
IS THIS POINT WITHIN LOADING REGION?

U: NO

P: ENTER ANOTHER DIAMETER.

U: 0.8

P: OPERATING POINT COORDINATES:
ABSCISSA= .049
ORDINATE= .075
IS THIS POINT WITHIN LOADING REGION?

U: YES

Having selected a diameter, the runstream proceeds at (c) below.

(b)

```
P: SPECIFY VAP. PRESS. DROP AT LOADING FROM FOLLOWING
   (1) 0.041,(2) 0.082,(3) 0.204,(4) 0.408,(5) 0.816,
   (6) 1.224 (KPA/M)
   NOTE: 0.408 (CODE 4) REPRESENTS APPROX. 60% OF FLOODING
   VALUE.
   ENTER APPROPRIATE CODE NO.

U: 4  (See section 7.332)

P: POLYNOMIAL OR ORDER 10 FITTED.
   SUM OF SQUARED ERRORS=    .0000589270

   OPERATING POINT ON SELECTED LOADING CURVE:
   ABSCISSA=      .039
   ORDINATE=      .056

   FOR GIVEN VAPOUR PRESSUR-DROP:
   SUPERFICIAL GAS VELOCITY= 4961.07  KG/HR.SQM
   RECOMMENDED TOWER DIAM.=      .33  M
   RESULTANT SUP. GAS VEL.=  4897.33  KG/HR.SQM
   RESULTANT SUP. LIQ VEL.=  3573.73  KG/HR.SQM
   ARE THESE FIGURES SATISFACTORY?

U: NO   (See section 7.333)
```

If the figures are rejected as shown above the runstream continues with:

```
P: ENTER NEW PRESSURE DROP CODE NO. FROM FOLLOWING
   (1) 0.041,(2) 0.082,(3) 0.204,(4) 0.408,(5) 0.816,
   (6) 1.224 (KPA/M)

U: 3

P: POLYNOMIAL OR ORDER 6 FITTED.
   SUM OF SQUARED ERRORS=   .0000086074

   OPERATING POINT ON SELECTED LOADING CURVE:
   ABSCISSA=     .039
   ORDINATE=     .033

   FOR GIVEN VAPOUR PRESSUR-DROP:
   SUPERFICIAL GAS VELOCITY= 3815.86  KG/HR.SQM
   RECOMMENDED TOWER DIAM.=      .38  M
   RESULTANT SUP. GAS VEL.=  3693.35  KG/HR.SQM
   RESULTANT SUP. LIQ VEL.=  2695.15  KG/HR.SQM
   ARE THESE FIGURES SATISFACTORY?
```

U: YES

If the results are acceptable the runstream continues at (d) below.

(c) The final output before the result table is a check on the maximum fowrates within the column.

```
P: FOR GIVEN VAPOUR PRESSURE-DROP:
   SUPERFICIAL GAS VELOCITY=  4698.14  KG/HR.SQM
   RECOMMENDED TOWER DIAM.=       .80  M
   RESULTANT SUP. GAS VEL.=   4474.42  KG/HP.SQM
   RESULTANT SUP. LIQ VEL.=   6638.07  KG/HR.SQM
   ARE THESE FIGURES SATISFACTORY?
```

U: YES : Calculations proceed - see (d) below

NO : (b) above is repeated until the flowrates are acceptable. (See section 7.333.)

(d) If no errors are encountered the next program output is the result table. (See section 7.35.)

7.33 Optional data inputs

All options relating to data required for the operating diagram are discussed in section 7.23 for Ponchon-Savarit designs. The only difference in the data input sequence for a packed column design is the absence of a Murphree tray efficiency factor.

7.331 Type and size of packing material: Data for four sizes of Raschig rings and three sizes of Berl saddles are included in the package. These data are stored in subroutine PAKING which is called by the mainline routine. They are: CF, E, AMF1, AMF2, ANF1A, ANF1B, ANF2A, ANF2B, PF1, PF2, DS, PHTW, PHSW, SWP and BETA, as defined in the nomenclature. Most of these factors are empirically derived and relate to Shulman's correlations [61]. The selection of packings included should be sufficient for tutorial purposes, but if users wish to consider other materials, the above factors for the particular packing must be derived or extracted from

literature. These can then be inserted in subroutine PAKING by extending the GO TO statement (line 3) to include another statement number, and listing the data at the end of the subroutine. This can then be accessed, without further alteration to the program, as follows:

P: DO YOU WANT TO USE A DIFFERENT TYPE OF PACKING?

U: YES

P: DO YOU HAVE THE PACKING DATA PREPARED?

(a) U: NO

P: REFER TO THE DISTILLATION PACKAGE MANUAL,SECTION 3.3 FOR FURTHER DETAILS.

The run is then terminated.

or (b)

U: YES

P: ENTER CODES FOR PACKING TYPE AND SIZE

U: 3,N where N is either a GO TO counter if more than one set of data is inserted in PAKING, or a dummy value (integer).

The above runstream is contained in subroutine PACTYP. Data presently included in PAKING are tabulated in Appendix A.

7.332 Column diameter calculations: Two options are available to the user:

(1) Specification of diameter in metres. (maximum 2 m)

If the user specifies the diamter, the program computes the resulting superficial vapour velocity, increases this value by 5% as a safety factor, and calculates the corresponding abscissa and ordinate values on the generalised flooding chart. (Figure 1, Appendix A.) The user then determines whether the point thus defined lies within the loading zone indicated on the chart. The resulting pressure drop

must also be checked at this stage, if a limit has been set on this value. Different values can be tried until a satisfactory design is obtained.

The program must be executed in demand mode in order to use this option effectively.

(2) Specification of allowable pressure drop.

An allowable pressure drop is determined firstly from flooding considerations and secondly in practice by economic analysis of the system.

(a) Flooding and loading.

In most industrial applications the upper limit of loading is 0,816 kPa/m (1,00 in H_2O/ft) and the lower limit is 0,204 kPa/m [17]. A pressure drop of 0,408 kPa/m represents operation at about 60% of flooding velocities and is often used in practical designs. Six curves are shown in Figure 1, Appendix A, and data for each are contained in elements FLOOD1 (0,041 kPa/m) to FLOOD6 (1,22 kPa/m). The user specifies a particular pressure drop corresponding to one of the six curves. The program fits a polynomial to the specified curve through subroutine FDATA, computes the abscissa and ordinate values, and determines the diameter of the column such that the resultant superficial vapour velocity is less than or equal to the value at the specified pressure drop. Diameters less than 2 m are checked at intervals of 0,01 m.

(b) Economic factors

In general, the lower the pressure drop, the lower the operating costs. For a given duty, however, the diameter of the column must increase in order to reduce the pressure drop, and thus the capital costs increase. An economic balance must be found and the appropriate pressure drop specified, bearing in mind flooding and loading considerations.

Other curves can be drawn on the chart representing specific pressure drop values if required. Data representing this curve must be compiled in a data element similar to those currently used by FDATA. These data elements are added to the runstream automatically through the ERTRAN facility. By calling the specific element FLOOD1, say, the specific curve may be considered without further alterations to the program. The user enters code number 1 to access FLOOD1. The FLOOD-data elements are set up as follows:

ND	No. of data pairs
A_k, B_k	k=1,ND (one pair per line).

A - abscissa; B - ordinate. Values extracted from Figure 1, Appendix A.

7.333 Vapour and liquid superficial flowrates: Before proceeding with the height of packing calculations the program prints out the superficial vapour and liquid flowrates resulting from the specified or calculated column diameter. This facility enables the user to investigate the effect of column diameter and pressure drop on the values of the flowrates and on each other.

7.334 Conversion factors: All conversion factors relating to the operating diagram are discussed in section 7.2. The following factors apply to mass transfer calculations performed in PACKED.

In the original work of Shulman [61], the F.P.S. system is used predominantly with respect to flowrates and densities. In deriving the correlations, however, a variety of units were used as shown in Appendix A, and conversion factors are required to maintain consistency. They are defined in DATA statements in PACKED.

ALBKGH	- lb ft^{-2} → kg m^{-2}
ASFSM	- ft → m
AMLCON,AMVCON	- viscosity values, see sections 7.345 and 7.346.
DLCON,DVCON	- diffusivity values, see sections 7.343 and 7.344.
RHOCON	- density values, see section 7.231.
SIGCON	- surface tension values, see section 7.342.
GC	- gravitational force constant (Figure 1, Appendix A) = 4,18 . 10^8 $lb_m/lb_f sec^2$

7.34 Physical Property Data Package

The following physical properties must be available to the packed column design program PACKED:

(a) Liquid density of each component and of the binary mixture.
(b) Surface tension of binary solution.
(c) Liquid-phase diffusivity.
(d) Vapour-phase diffusivity.
(e) Liquid viscosity of binary solution.
(f) Vapour viscosity of binary solution.
(g) Density of water.

Physical property data tables in the literature are often sparse, or non-existent, for many substances. To facilitate collation of data for setting up tutorial problems, or studying real situations, a number of procedures for computing each of the above properties are available to the user.

Data required by each routine is added to the runstream, through the UNIVAC ERTRAN facility. All data elements must be set up in free format. Subscript 1 refers to the light component in the following discussion.

7.341 Liquid Density. Subroutine RHOLIQ: Both pure-component and solution density data, as functions of temperature, are required by PACKED. At least one temperature-density data pair for each pure component must be available. Perry [58] is a good source of density values, usually quoted at 20° or 25°C. Pure component densities are computed in one of two ways for each substance of component density specified at some temperature T_1.

(1) Single values are extrapolated to higher temperatures T_2 using Goldhammer's correlation [67]:

$$\frac{[\rho_L - \rho_V]_2}{[\rho_L - \rho_V]_1} = \left[\frac{T_C - T_2}{T_C - T_1}\right]^{EXP}$$

where EXP takes the following values according to the type of compound considered.

Compound	EXP
Alcohols and Water	0,25
Saturated, Unsaturated and Aromatic Hydrocarbons and Ethers	0,29
Other Organic compounds	0,31
Inorganic compounds	0,33

Vapour density ρ_V is neglected, which assumes that systems considered will be at pressures less than 10^3 kPa.

(2) Temperature-density data tables, for a few compounds, can be found in the literature [58]. If such a table is available for either component, the routine will determine the component density at any temperature, within the range of the table, using linear interpolation between the data. The temperature range of the data table should span the column temperature range because end-values are taken for column temperatures falling outside the table limits.

Either of the above procedures may be used for either component. The subroutine selects the appropriate method on the basis of data supplied.

Data elements must be compiled as follows:

(a) For all systems..

(1) Element ROLDA1

AMW_1, TC_1	Molecular weight and critical
AMW_2, TC_2	temperature data (in ^{o}C) for each component.
NDAT1,NDAT2	Number of temperature-density data points available for each component.

(2) Element ROLDA2

T_1, ρ_1	NDAT1 data sets for component 1,
T_1, ρ_1	in order of increasing temperature (in ^{o}C).
T_2, ρ_2	NDAT2 data sets for component 2,
T_2, ρ_2	in order of increasing temperature (in ^{o}C).

Note: (i) Conversion factor RHOCON in mainline program PACKED, converts the units in which the density data are specified here, to lb ft^{-3} for use in the mass-transfer correlations. The present value of RHOCON assumes that density data are specified in g $m\ell^{-1}$.

(ii) RHOLIQ is called by VISCL and SURTEN to provide liquid-density data for specific correlation expressions in these subroutines. In each case it is assumed that density data are in units of g $m\ell^{-1}$. If data are supplied to RHOLIQ in units other than g $m\ell^{-1}$, these routines and factor RHOCON must be adjusted accordingly.

(iii) NDAT1 and NDAT2 can take any value ≥ 1 and need not be equal. For linear interpolation the accuracy depends on the number of data pairs available.

(b) For systems for which composition-density data are available.

(3) Element ROLDA3

NDATA,TBASEC	Number of data pairs and base temperature ^{o}C
XS,RHOS	NDATA composition-density data pairs in order of incrasing composition.
XS,RHOS	XS refers to the light component.

Note: Density data units here must be the same as those used in ROLDA2, so that the conversion factor, RHOCON, applies correctly.

7.342 Surface Tension: Subroutine SURTEN: The surface tension of the binary solution, in dyne cm^{-1}, is required by the mainline program, for computing liquid holdup factors.

The surface tension of the mixture is determined either directly from a composition-surface tension data table by linear interpolation or from pure-component surface tension values, using Stakhorsky's expression[69] (see section 4.552, Equation (4.22)). Temperature extrapolation of single values is performed using Lovgren's correlation. (See section 4.552, Equation (4.25).)

Pure-component surface-tension data may be provided as a single temperature-surface tension pair, a table of such pairs for a range of temperature sufficient to cover the column temperature profile, or as molecular weight and parachor data from which the surface tension is computed.

If no pure-component or solution, surface tension data can be found in the literature the Sugden parachor must be

computed as shown in Perry [58].

Data elements must be compiled as follows:

(a) For all systems.

Element SIGDA1

1. NSIGM — Counter specifying availability of composition-surface tension data for mixture.
   NSIGM=1 - pure component data to be used.
   NSIGM=2 - solution data available.
2. $AMW_1, TC_1, PARA_1$
3. $AMW_2, TC_2, PARA_2$
4. NDAT1, NDAT2

TC in ^{o}C, PARA 1 and 2 - Sugden parachors for each component. (See Perry [58] Table 3.317, p.3-240.)

If NSIGM=2 all values above, except TC_1 and TC_2 can be zeros. If temperature-surface tension data is to be used for either component NDAT1, NDAT2 specifies the number of data pairs available for each component. If parachor values are used for either component the corresponding NDAT is set to zero. If parachors are to be used the molecular weights must also be provided. If either parachor is not used it must be set to -1.

Elements SIGDA2 and SIGDA4

These elements are only required if NDAT1 or NDAT2 are greater than 0.

(a) SIGDA2

$T_{1_k}, SIG1_k$ k=1,NDAT1 T in ^{o}C in order of increasing T. SIG1 in dyne cm^{-1}

(b) SIGDA4

$T_{2_k}, SIG2_k$ k=1,NDAT2 As above.

Element SIGDA3

This element is only required if NSIGM=2 in SIGDA1.

NDATA,TBASEC	Analogous to ROLDA3 described previously; SIGS in dyne cm^{-1}
XS_k,$SIGS_k$	k=1,NDATA

Note: Conversion factor SIGCON in the mainline program converts values supplied by SURTEN to units of dyne cm^{-1}. The present value (1,0) assumes that data used by SURTEN is in the correct units.

7.343 Liquid Diffusivity: Subroutine DIFLIQ: Liquid diffusivity data are tabulated in Perry [58] for a few binary systems. The subroutine requires a single diffusivity value and the temperature at which it is quoted. Diffusivities are usually quoted as the diffusivity of component A in component B at infinite dilution of component A. For the purposes of the subroutine DIFLIQ, the light component is component 1. The subroutine uses these data to determine the diffusivity of a solution for which the temperature and composition is provided by the calling routine. If the diffusivity of a system can not be found in the literature, a value should be calculated using an expression such as that of Wilke and Chang [71].

Diffusivity must be specified in cm sec^{-1}. A conversion factor in the mainline program, DLCON, converts the temperature corrected value to units of ft^2hr^{-1} for use in the mass transfer calculations. The temperature at which the diffusivity value is determined, is supplied in ^oC.

Data element required:

Element DLDAT

DL12,TDL - a single diffusivity-temperature data pair in cm^2sec^{-1} and ^oC respectively. DL12 refers to the diffusivity of the light component at infinite dilution in the heavy component.

Note: (i) DL12 above is equivalent to $D_L \times 10^5$. The factor 10^{-5} is built into the subroutine because most liquid diffusivities fall within the range $0,5 \times 10^{-5} - 5,0 \times 10^{-5}$. For example, if the diffusivity of a system, at 25°C, is computed to be $2,6 \times 10^{-5}$, the values 2.6,25.0 are inserted in the data element DLDAT.

(ii) Present value DLCON=3,875, in mainline program PACKED assumes that data used by DIFLIQ are in cm^2sec^{-1}.

7.344 Vapour Diffusivities: Subroutine DIFVAP: Vapour phase diffusivities, required in PACKED for computing Schmidt numbers are calculated here using Arnold's [72] correlation. The molecular weight, molar volume and normal boiling point of each component must be specified.

The diffusivity calculated using this correlation has units of cm^2sec^{-1}. Conversion factor, DVCON, in the mainline converts the value to units of ft^2hr^{-1} where necessary. The molecular weight, molal volume and normal boiling point for each component must be known.

Data element required:

Element DVDAT

AMW_1, VB_1, TB	Molecular weight, molal volume,
AMW_2, VB_2, TB	and boiling point (°C), specified for each component

Note: (i) Molal volume VB in cm^3gmole^{-1} may be computed as $MW.\rho_{L_{bp}}^{-1}$, where $\rho_{L_{bp}}$ is the component liquid density at its normal boiling point or by summing contributive atomic volumes as tabulated in Perry [58] Table 3-306, p.3-229.

(ii) DVCON=DLCON in mainline. (See section 7.343.)

7.345 Liquid Viscosity: Subroutine VISCL: Two procedures for determining these data, at given temperatures, are included in VISCL.

The first technique estimate component liquid-viscosities using Thomas's equation, which is considered to be the best generalised expression available [75]. This correlation requires the critical temperature and viscosity constant of each component.

$$\mu_L = 0{,}1167\rho_L^{0{,}5}10^{\gamma}$$

where $\gamma = B(1-T_r)/T_r$

The viscosity constant, B, is calculated by summing contributory atomic terms tabulated in Perry [58] Table 3-321. This expression may be applied satisfactorily for reduced temperatures, $T_r \leq 0{,}7$.

Units of the calculated viscosity value are centipoise. Conversion factor AMLCON is the mainline, converts the units to lb $ft^{-1}hr^{-1}$ as required.

The alternative procedure for computing liquid viscosity data requires the user to provide a temperature-viscosity data table, the temperature range of which should cover the column temperature-profile.

Liquid mixture viscosity is calculated using the component data computed above and the Kendall-Monroe expression [74].

Data elements required.

Element FMUDA1

TC_1, B_1	Critical temperature data, TC_1,
TC_2, B_2	supplied in ^{o}C. If B factors are not used, zeros must be inserted in the element.
NDAT1,NDAT2	NDAT1,NDAT2 are the numbers of data pairs, if any, to be supplied in FMUDA2 and FMUDA3 respectively.

Element FMUDA2

T_{1_k},$AMU1_k$ k=1,NDAT1. FMUDA2 is required only if $B_1 \leq 0$. Viscosity data in centipoise.

Element FMUDA3

T_{2_k},$AMU2_k$ k=1,NDAT2. FMUDA3 is required only if $B_2 \leq 0$. Viscosity data in centipoise.

7.346 Vapour Viscosity: Subroutine VISCV: Vapour viscosity data are calculated in a manner similar to that used in VISCL. The same options are available to the user, although the correlation for pure-component data is different. Arnold's relationship is used [77].

The molecular weight, molar volume and normal boiling point of each component must be specified as for DIFVAP. The computed vapour viscosity is in units of micropoise. Conversion factor AMVCON converts the values to units of lb ft^{-1} hr^{-1} in the mainline program PACKED, where necessary.

Vapour-mixture viscosity data are calculated using the relationship of Herning and Zipperer [76].

If tables of temperature-vapour-viscosity data are available, linear interpolation is used to determine required values.

Data elements required.

Element VMUDA1

AMW_1,VB_1,TB_1

AMW_2,VB_2,TB_2

NDAT1,NDAT2

NDAT1,NDAT2 are zero if data tables are not to be used.

Elements VMUDA2,VMUDA3

These data elements are analogous to FMUDA2 and FMUDA3 and need be set up only if data tables are available. If they are used the data are inserted as T_k,AMU_k with the table set out in order of increasing temperature values. Vapour viscosity data must be supplied in micropoise, or factor AMVCON in PACKED must be altered so that the units are converted to lb $ft^{-1}hr^{-1}$.

7.347 Density of Water: Subroutine H2ORHO: Data element H2ORHO supplies a temperature-density data table for water, which need not be altered by the user. Data in this table are in units of ^{o}C and g $m\ell^{-1}$ respectively. To expand the table, or change the units the format must be as follows:

Data element H2ORHO

NDATA	Number of data pairs in table.
T_k,ρW_k	k=1,NDATA (in order of increasing temperature.)

7.35 Result Table

The format of the result table is similar to that for Ponchon-Savarit design of tray columns. Column diameter and heights of packing required in each section are included. Two data tables may be suppressed if not required, to save output time on the terminal. These are

(i) P: DO YOU WANT COMPOSITION,TEMPERATURE AND FLOWRATE DATA TABLES?

U: YES or NO

(ii) P: DO YOU WANT MASS-TRANSFER DATA TABLES?

U: YES or NO

The tables produced in (i) and (ii) above are similar to those shown in Treybal [9] pp. 365 and 367 respectively.

No plotting routine is associated with PACKED.

7.36 Error Messages

Origin	Error Message	Description
PACKED	PINCH POINT OCCURS AT POINT...,... ON THE EQUILIBRIUM CURVE. THE REFLUX RATIO OR THE DISTILLATE AND BOTTOMS COMPOSITIONS SHOULD BE ADJUSTED ACCORDINGLY.	Pinch point detected.
	DIAMETER OF THE COLUMN EXCEEDS 2 METRES SYSTEM UNSUITABLE FOR PACKED COLUMN DISTILLATION	Limit of 2 metres suggested by various workers [59,60,61].
FDATA	ERROR IN MATRIX. JC=...	Error in UNIVAC routine GJR. Consult UNIVAC MATHPAC manual.

Other error messages occurring in PACKED are described in section 7.26.

7.37 Dimension parameters and specification limitations

(a) Dimension parameters

There are four parameters associated with PACKED. The COMMON variables are the same as those used in BINARY and consequently the same parameters are used for these. They are NPEQ,NPEN and NPPS. Certain variables appear only in PACKED and use the parameter NPAK which must be equal to NPPS. A separate variable name is used for easy reference when scanning the COMMON block.

(b) Specification limitation

The same limitations described in section 7.2 apply here. Most of the physical property correlations included in the package are valid only for systems at less than 10^3 kPa pressure. The column diameter is limited to a maximum of 2 m [59,60,62].

7.4 MULTICOMPONENT DISTILLATION

7.41 Application

The multicomponent distillation program MULCOM is a simulation package based on the calculation procedure proposed by Thiele and Geddes [2]. The θ method of convergence, proposed by Lyster et al. [3] is used to obtain a converged solution.

The program can simulate columns with one feed stream and no side streams, using a partial reboiler with either a total or partial condenser. The user may either specify equimolal overflow conditions, or consider the effects of stream enthalpy. Tray efficiency can be included through vaporization efficiency data for each component. For any system the following data must be known or estimated:

(1) Number of components in the feed stream.
(2) Number of trays in the column.
(3) Position of feed tray.
(4) Condenser operating pressure.
(5) Pressure drop per tray and pressure drop across the condenser.
(6) Reboiler and condenser temperature.
(7) Feed rate, composition, temperature and pressure.
(8) Feed enthalpy if the pressure of the feed is greater than the pressure on the feed tray.
(9) Type of condenser (total or partial).
(10) Enthalpy and equilibrium data (see section 7.43 for details).
(11) Vaporization efficiency data if ideal trays are not assumed.
(12) Reflux ratio and distillate product withdrawal rate.

The program is designed to be run in demand mode, with interaction between the program and the user. It can also

be run in batch mode providing the runstream is set up as for demand operation.

Subroutines used by MULCOM include:

Subroutine	Purpose
TEACH	Provides a brief description of the program's capabilities.
READBD	Reads all basic problem data, except that required by subroutine ENTHAL.
EQUCON	Computes equilibrium constants for each component on each tray.
QFLASH	Isobaric or isenthalpic flash calculation.
ENERGY	Standard enthalpy calculation routine.
ENTHAL	Optional enthalpy routine supplied by the user. (See section 7.45.)
BUBCAL,DEWCAL	Bubble point and dew point calculations respectively.
FLOWEQ	Flowrate profiles though column assuming McCabe-Thiele conditions.
FLOWNE	Flowrate profiles considering enthalpy effects.
ABSTRP	Absorption and stripping factor calculations.

The following six subroutines perform the Thiele-Geddes tray-to-tray calculations.

VBLV,VDLD	Material balance and equilibrium calculations.
BDCALC,BDCORR	Feed tray matching.
VLRECT,VLSTRP	Stream compositions.
THECAL	Convergence procedure: calculation of θ in Holland's θ method.
RUNRES	Abbreviated table of results.
OUTPUT	Detailed table of results showing composition and molar flowrates of components in each stream.

The mainline program serves to link the subroutines together and to direct the overall calculation procedure.

In order to minimise core requirements, MULCOM must be mapped as follows:

```
@MAP DISTFILE.MULCOMM,.MULCOM
```

where element MULCOMM contains the following statements:

```
 1          SEG MULCOM1
 2          IN MULCOM
 3          SEG READINS*,(MULCOM1)
 4          IN READIN
 5          SEG QFLASHS*,(MULCOM1)
 6          IN QFLASH
 7          SEG FLOWEQS*,(MULCOM1)
 8          IN FLOWEQ
 9          SEG FLOWNES*,(MULCOM1)
10          IN FLOWNE
11          SEG ABSTRPS*,(MULCOM1)
12          IN ABSTRP
13          SEG VBLBS*,(MULCOM1)
14          IN VBLB
15          SEG VDLDS*,(MULCOM1)
16          IN VDLD
17          SEG BDCALCS*,(MULCOM1)
18          IN BDCALC
19          SEG THECALS*,(MULCOM1)
20          IN THECAL
21          SEG BDCORRS*,(MULCOM1)
22          IN BDCORR
23          SEG VLSTRPS*,(MULCOM1)
24          IN VLSTRP
25          SEG VLRECTS*,(MULCOM1)
26          IN VLRECT
27          SEG RUNRESS*,(MULCOM1)
28          IN RUNRES
29          SEG OUTPUTS*,(MULCOM1)
30          IN OUTPUT
31          LIB DISTFILE.
```

7.42 Operations of the program

7.421 Graphical representation of results: The plotting program PLOTTER included in the package (see section 7.5) is able to construct the following diagrams describing a system simulated by MULCOM.

(a) Temperature profile.
(b) Liquid and Vapour flowrate profiles.
(c) Distillate composition bar diagram.
(d) Component composition profiles.

Within a single execution of MULCOM it is possible to simulate a system several times for different values of key variables. Results for a maximum of five runs may be stored on file for subsequent use by the plotting program. On executing PLOTTER the user must specify the run for which a full plot is required. PLOTTER then selects the specified set of data from the data files and constructs diagrams (b), (c) and (d) above. Diagram (a) will automatically superimpose the temperature profiles for all the runs stored on file, on a single set of axes. These points are discussed in detail in section 7.52.

Files for data storage are assigned by MULCOM during the course of the run, if the user indicates that these are required. (See section 7.435.)

7.422 Execution of MULCOM in demand mode with conversational interaction: The following instruction is entered to start the run:

```
@ADD DISTFILE.STARTMC
```

STARTMC contains the following instructions:

```
@ASG,UP PRINTFILE.
@USE 16,PRINTFILE.
@XQT DISTFILE.MULCOM
```

PRINTFILE is assigned to the run for storing detailed results tables than can be sent to the line printer on termination of the run. (See section 7.44.) This file is not used by the plotting program.

The run then proceeds as follows. All instructions and data are entered in free format and it is suggested that bulky tables, such as equilibrium or enthalpy data, be inserted in data-elements for addition to the runstream as required.

The program begins with the standard format.

```
P: ANSWER ALL QUESTIONS YES OR NO
   ALL INPUT IN FREE FORMAT.

   DO YOU KNOW HOW TO USE THIS PROGRAM?
```

If the user response is NO the following description is printed out and the run is terminated.

```
P: THIS IS A MULTICOMPONENT DISTILLATION SIMULATION
   PROGRAM.
   THE CALCULATION PROCEDURE USED IS THAT FIRST PROPOSED
   BY THIELE AND GEDDES,COMBINED WITH HOLLANDS THETA
   METHOD OF CONVERGENCE.
   THE PROGRAM IS LIMITED TO SYSTEMS WITH ONE FEED AND
   NO SIDE STREAMS.

   REFER TO THE DISTILLATION PACKAGE MANUAL,SECTION 4,
   FOR FURTHER INFORMATION.
```

For a response of YES, the run continues with

```
P: DO YOU WANT TO ADD A DATA ELEMENT?
```

As for other programs, a response of YES requires the addition of a previously prepared data element. A sample element is shown in Table 7.41.

If the conversational facility is required the runstream proceeds as follows. Only the program output is shown. Data is entered in free format or by addition of data-elements where options are available or further explanation is required, reference is made to a subsequent section.

TABLE 7.41

PROBLEM-DATA ELEMENTS FOR EXECUTION OF MULCOM

1	6	No.of components
2	14	*No.of trays
3	9	*Feed tray number
4	150.0,3	Condenser pressure and code.
5	10.0,1	Tray pressure drop and code.
6	25.0,1	Condenser pressure drop and code
7	236.0,2,115.0,2	Estimate of reboiler and condenser temperatures with corresponding codes.
8	METHANE	Component names
9	ETHANE	
10	N-PROPANE	
11	N-BUTANE	
12	N-PENTANE	
13	N-HEXANE	
14	100.0	FTOT
15	0.03,0.07,0.15,0.33, 0.30,0.12	Z_i i=1,NC
16	82.2	T_F
17	YES	'ARE ENTHALPY EFFECTS TO BE CONSIDERED?'
18	NO	'IS FEED PRESSURE COLUMN PRESSURE?'
19	PAR	Partial condenser specified.
20	YES	**'DO YOU WANT TO USE YOUR OWN ENTHALPY ROUTINE?'
21	1	Antoine constants specified.
22	6.61184,389.93,266.0	Antoine constants
23	6.80266,656.40,256.0	
24	6.82973,813.20,248.0	
25	6.83029,945.9,240.0	
26	6.85221,1064.63,232.0	
27	6.87776,1171.53,224.366	
28	YES	Convergence limit specification
29	0.0001	
30	NO	Efficiency data specification (See section 7.434.)
31	2	
32	0.91,0.92,0.95,0.98, 1.05,1.09	
33	1.2,38.46	R,DTOT

* Trays are numbered from the bottom, excluding the reboiler and condenser

** See Table 7.42.

P: ENTER NO. OF COMPONENTS IN FEED.

U: integer value

P: ENTER NO. OF TRAYS IN COLUMN
EXCLUDE CONDENSER AND REBOILER

U: integer value

P: ENTER FEED TRAY POSN. (NO.FROM BOTTOM,REBOILER=0)

U: integer value

P: ACCEPTED PRESSURE UNITS: 1=MMHG,2=PSIG,3=PSIA,4=KPA,
5=ATM.
ENTER CONDENSER PRESSURE AND UNIT CODE. (EG. 760.0,1)

P: ACCEPTED PRESSURE DROP UNITS: 1=MMHG,2=PSI,3=KPA
ENTER PRESSURE DROP PER TRAY AND UNIT CODE.

P: ENTER PRESSURE DROP ACROSS CONDENSER AND UNIT CODE.

P: ACCEPTED TEMPERATURE UNITS: 1=DEG C,2=DEG F
ENTER ESTIMATE OF REBOILER AND CONDENSER TEMPERATURES
WITH UNIT CODES. (EG. 100.0,1,50.0,2)

The pressure inputs have been discussed in section 7.1, the temperature input format is shown. The example represents:

reboiler temp. 100°C
condenser temp. 50°F

P: ENTER NAMES OF COMPONENTS. (ONE PER LINE,MAX 30
CHARACTERS EACH.)

P: ENTER FEED RATE. (KG MOLES/HR.)

P: ENTER FEED STREAM COMPOSITIONS. (MOL FRACTIONS)

P: ENTER FEED TEMPERATURE. (DEG C)

Component names entered as shown in section 7.1. The feed stream data are entered as real numbers in free format.

P: ARE ENTHALPY EFFECTS TO BE CONSIDERED?

U: YES - for options see section 7.431

or NO

P: TOTAL OR PARTIAL CONDENSER?
ENTER TOT OR PAR.

U: TOT or PAR

At this point enthalpy data are required, if these effects are considered (see section 7.432). Once these data have been inserted, or if enthalpy is not considered, the run proceeds as follows:

P: ENTER EQUIL. DATA AS DESCRIBED IN THE MANUAL,SECT.4, OR ADD A DATA ELEMENT.

For options see section 7.433.

P: DO YOU WANT TO SPECIFY THE THETA CONVERGENCE LIMIT?

U: YES

P: ENTER CONVERGENCE LIMIT.

U: real value (e.g. 0.0001)

This is the limit set on the θ convergence procedure. If the limit is not supplied by the user a value of 0.00005 is adopted.

P: ARE PERFECT TRAYS ASSUMED?

For options, see section 7.434.

P: ENTER REFLUX RATIO AND DISTILLATE WITHDRAWAL RATE.

U: real values - reflux ratio, distillate rate.

P: DO YOU WANT GRAPHICAL OUTPUT? (SEE MANUAL SECT.4)

For options, see section 7.435.

This completes the input of basic data required to solve a problem. The run is then started and identified through the following two statements.

P: TO PROCEED,TYPE GO OR TO STOP THE PROGRAM,TYPE END.

U: GO

P: ENTER RUN NO. AS: NO.,DAY,MONTH. (INTEGER VALUES)

U: run identification (e.g. 1,4,10)

The acceptance of all data by the program is acknowledged by the statement:

P: START OF COMPUTATION.

Values of θ are printed out with each iteration, enabling the user to follow the course of the convergence. On completion of the calculations an abbreviated table of results is printed on the terminal. The runstream then continues:

P: DO YOU WANT A FULL PRINT-OUT?

On the basis of the printed result table, the user may require a full set of results for the run. These are written to the previously assigned file PRINTFILE. (See section 7.44.) The program then continues with:

P: TO PROCEED,TYPE GO OR TO STOP THE PROGRAM,TYPE END.

The user may now terminate the run or solve the problem again for different values of certain selected variables shown below.

U: GO

P: ENTER RUN NO. AS: NO.,DAY,MONTH. (INTEGER VALUES)

U: new run identification

P: ENTER REFLUX RATIO AND DISTILLATE RATE FOR NEW RUN.

P: ENTER FEED TEMPERATURE. (DEG C)

If enthalpy data was supplied originally the feed stream pressure may be specified higher than the column pressure, and its enthalpy supplied, as discussed in section 7.431. The final variable that may be altered is:

P: ENTER FEED TRAY POSN. (NO. FROM BOTTOM,REBOILER=0)

The computation then proceeds as before. Any number of runs may be performed in this manner within a single execution, but a maximum of five full result tables may be requested. This limit is to discourage the generation of excessive amounts of printout.

The program terminates as follows:

P: TO PROCEED,TYPE GO OR TO STOP THE PROGRAM,TYPE END.

U: END

P: SUCCESSFUL EXECUTION COMPLETED.

If full result tables have been stored a set of instructions for sending these to the line printer is also printed out here.

7.43 Input options

7.431 Feed conditions: The feed may be specified as being at the same pressure as the feed tray, or at a higher pressure. A different flash calculation procedure is used in each case. If the pressure of the feed is above that on the feed tray, the flash calculation is isenthalpic and the feed enthalpy must be specified in kJ/kg mole. This procedure obviates the need to provide enthalpy data at both the column pressure and the feed pressure.

The program responds as follows if the user considers enthalpy effects:

P: IS FEED PRESSURE GREATER THAN COLUMN PRESSURE?

U: NO or YES

P: ENTER FEED ENTHALPY. (KJ/KGMOLE)

U: real value

The main runstream continues, as shown in section 7.422, with

P: TOTAL OR PARTIAL CONDENSER?

7.432 Enthalpy data: There are two procedures available for including enthalpy effects in problems:

(a) Library routine:

The user may use one of the functions built into the library enthalpy routine. These functions were chosen because they have been used by a number of workers [4,18,21, 24,25] and thus appropriate coefficients for several problems are readily available. These are three functions available:

(1) $(H_{i,j})^{\frac{1}{2}} = a_i + b_i T + c_i T^2$ T_j in ^{o}R (NENTH=1) (7.4.1)

(2) $H_{i,j} = a_i + b_i T + c_i T^2$ T_j in ^{o}F (NENTH=2) (7.4.2)

(3) $H_{i,j} = a_i + b_i T + c_i T^2$ T_j in ^{o}R (NENTH=3) (7.4.3)

where $T=T_j/100$ and T_j is the temperature on stage j. Similar equations define the liquid enthalpy $h_{i,j}$.

Stream enthalpies are computed as

$$H_j = \Sigma y_{i,j} H_{i,j} \qquad (7.4.4)$$

and

$$h_j = \Sigma x_{i,j} h_{i,j} \qquad (7.4.5)$$

The coefficients for the above functions quoted in the literature are derived from enthalpy data in Btu/lb mole. An informal conversion factor ENCONV converts all calculated enthalpies to SI units of kJ/kg mole. Another factor CONFAC converts heat duties from Btu/hr to kilowatts. If coefficients cannot be found they must be derived by fitting one of the two above functions to available data.

It should be noted that the functions are written in terms of $T_j/100$ rather than T_j. This facilitates free format data input in the following way:

Suppose a set of coefficients for one of the functions have been derived or extracted from the literature in terms of T_j. A typical set could be: a_i = 687,0; b_i = 0,9964;

c_i = 1,9395 x 10^{-4}. These would be supplied to the program, in free format, as follows:

U: 687.0,99.64,1.9395

The factor $T_j/100$ will set the decimal point correctly. This procedure was proposed by Holland [21].

(b) User routine:

The user may provide an enthalpy routine incorporating any convenient procedure for computing enthalpies. The routine must conform to certain specifications outlined in section 7.45 which ensure that the routine provides a facility equivalent to the library routine. An example of such a routine, incorporating Kobe's heat capacity functions, is included in the package. (See section 7.45.)

The routine reads in its own data using the UNIVAC ERTRAN facility, and communicates with the calling program through the calling list. Data input is thus not considered in the run-stream.

The derivation of coefficients for the library functions is a time consuming process. It is often more convenient when setting up tutorial problems to use available data directly or to employ some expression for which coefficients are known.

The program determines the procedure required by the user as follows:

P: DO YOU WANT TO USE YOUR OWN ENTHALPY ROUTINE?

U: NO or YES

P: IT IS ASSUMED THAT THE ROUTINE CONFORMS TO SPECIFICATIONS SET OUT IN SECT. 4.5 IN THE MANUAL.

* (see below)

P: REFER TO MANUAL,SECT. 4.32,FOR ENTHALPY ROUTINE CODES. ENTER CODE NO. OF LIBRARY ENTHALPY ROUTINE REQUIRED.

U: integer value (NENTH) corresponding to selected function as shown above.

P: ENTER ENTHALPY DATA AS DESCRIBED IN THE MANUAL, SECT. 4.32 OR ADD A DATA ELEMENT.

For a sample enthalpy data block, see Table 7.42.

* (see below)

Coefficients for the chosen function must be supplied as follows:

(a) One set of a_i,b_i,c_i per line
(b) In order from light component to heavy component.
(c) Liquid enthalpy coefficients for all components first, then all vapour enthalpy coefficients.

It is often more convenient to add these data in a block data element.

* The program then continues with the main runstream. (See section 7.422.)

TABLE 7.42

ENTHALPY DATA FOR LIBRARY ROUTINE ENERGY

The following data are extracted from Holland [21] and correspond to NENTH=2, for a five component system.

```
 1      0.,2521.92,175.417      ]
 2      0.,3345.,150.           |
 3      0.,2960.,400.           | Coefficients for liquid
 4      0.,3681.33,283.334      | enthalpy function a_i,b_i,c_i
 5      0.,3845.,250.           ]
 6      7488.65,1750.51,79.273  ]
 7      9592.01,1843.56,221.433 |
 8      8002.89,4382.70,-401.587| Coefficients for vapour
 9      11645.77,2770.55,156.345| enthalpy function a_i,b_i,c_i
10      12004.88,3168.66,67.456 ]
```

7.433 Equilibrium data: Four functions are available in subroutine EQUCON for computing vapour-liquid equilibrium constants.

NEQUIL=1 (1) Antoine's Equation combined with Raoult's and Henry's Laws.

NEQUIL=2 (2) Equilibrium constant expressed as a function of temperature.

NEQUIL=3 (3) Relative volatility and a key component equilibrium constant expressed as function of temperature, with the remaining equilibrium constants determined as the product of these.

NEQUIL=4 (4) Equilibrium constant expressed as another function of temperature.

The latter three techniques have been included for the same reasons as the library enthalpy routine. If coefficients for these are not available, their derivation is tedious and time consuming. Antoine constant by comparison are readily available for a wide variety of compounds. The use of this equation is, however, highly idealised.

Data for each function must be supplied as follows. Coefficients must be adjusted for functions in terms of T/100 or T/1000 as specified below.

(1) Antoine's Equation

Calculation procedure:

(a) From Antoine's equation

$$P_{V_{i,j}} = 10^{[A_i+B_i/(C_i+T_j)]} \quad T_j \text{ in } {}^{o}C \qquad (7.4.6)$$

From Raoult's law

$$y_{i,j} = P_{V_{i,j}} x_{i,j}/P_T$$

From Henry's law

$$y_{i,j} = K_{i,j} x_{i,j}$$

(b) $K_{i,j} = P_{V_{i,j}}/P_T$ (7.4.7)

Data in free format, preferably in a data element (see Table 7.41):

1 (NEQUIL)

A_i, B_i, C_i i=1,NC (Antoine constants, one set per line light-heavy component order)

(2) Function of Temperature

This function has been used by Seppala and Luus [18]

$K_{i,j} = \exp[a_i - (b_i/T) + c_i T]$ $T=Tj/100$, T_j in ^oR (7.4.8)

Data in free format, coefficients adjusted for $T=T_j/100$.

2 (NEQUIL)

a_i, b_i, c_i, O^* i=1,NC ($K_{i,j}$ coefficients, one set per line, light-heavy component order)

(3) Relative volatility and key component K value

Calculation procedure:

$K_{b_k} = a + bT + cT^2 + dT^3$ (7.4.9a)

$\alpha_{i,j} = e_i + f_i T + g_i T^2$ $T=T_j/100, T_j$ in ^oF (7.4.9b)

and $K_{i,j} = \alpha_{i,j} . K_{B_k}$ where k signifies the base component.

Data in free format, coefficients adjusted for $T=T_j/100$

3 (NEQUIL)

a,b,c,d (K_{b_k} coefficients)

e_i, f_i, g_i, O^* i=1,NC ($\alpha_{i,j}$ coefficients, one set per line, light-heavy component order)

Note: $\alpha_{i,k}$ coefficients are 1.,0.,0.,0.

(4) Function of Temperature

This function has been used by Holland [21]

$$K_{i,j} = T_j \left[a_i + b_i T + c_i T^2 + d_i T^3\right]^3 \quad T=T_j/1000 \quad (7.4.10)$$

T_j in ^{o}R

Data in free format, coefficients adjusted for $T=T_j/1000$

4 (NEQUIL)

a_i, b_i, c_i, d_i* i=1,NC ($K_{i,j}$ coefficients, one set per line, light-heavy component order)

*The same READ statement is used for all three sets of coefficients. The fourth array must therefore be filled in each case with real values or zeros as required.

The program then continues with the main runstream.

7.434 Tray efficiency: Tray efficiency is considered in terms of component vaporization efficiency factors defined as

$$E_{i,j} = \frac{y_{i,j}}{K_{i,j} x_{i,j}} \quad (7.4.11)$$

These factors should be extracted from literature or determined experimentally for simulating real systems. Vaporization efficiency factors have not been determined for many systems, however, and the experimental procedure for obtaining realistic data is extremely complex.

Efficiency factors may be estimated in order to demonstrate the effects of tray performance on a system, but this procedure can not be used to describe real columns in practice [22]. Data may also be derived from real and ideal equilibrium data using the above equation.

If the user's reply to the main runstream question is NO the program responds:

P: ADD EFFICIENCY DATA AS DESCRIBED IN THE MANUAL.

Data may be supplied in two ways, preferably through data element:

(1) Efficiency factors known for each component on each tray

1 (counter MEFF defining this option)

$E_{1,NP1}, E_{2,NP1}, \cdots E_{NC,NP1}$ (Condenser stage, factors for all components from light to heavy on one line)

.

.

.

$E_{1,0}, E_{2,0}, \cdots E_{NC,0}$ (Reboiler stage)

(2) Efficiency factors known for each component on one stage; assumed equal on all stages.

This is not the case in practice, but is useful for demonstrating the basic concept of efficiency. (See Table 7.41)

2 (counter MEFF defines this option)

$E_{1,n}, E_{2,n}, \cdots E_{NC,n}$ (Efficiency factors for each component on stage n)

Suitable tables of data for a number of systems are given in the literature [23,41,44,45].

The program continues with the main runstream after reading this data. If perfect trays are assumed, efficiency data are not required, and the factors are set to 1,0. This section is then bypassed.

7.435 Data storage for graphical representation: If the graphs described in section 7.421 are required, files must be assigned for storage of data. A maximum of five data sets may be stored during a single execution. (See section 7.44.)

The runstream proceeds as follows if the user replies YES to the main runstream question:

P: DO YOU KNOW HOW TO USE THE PLOT ROUTINE?

U: YES or NO → return to main runstream.

↓

P: ENTER DATA-FILE NAMES.

Four files are required and the name of each must be six characters in length. File names are typed in on one line, spaced with commas.

U: DFILE1,DFILE2,DFILE3,DFILE4

These are then automatically assigned by the program through the ERTRAN facility. The program returns to the main runstream.

7.436 Conversion factors: Conversion factors CONFAC and ENCONV are defined in the enthalpy subroutine ENERGY and are discussed in section 7.235. No other conversion factors are used by the standard program MULCOM. Factors required by a user enthalpy subroutine must be included in the subroutine itself. (See section 7.45.)

7.44 Program result tables

An abbreviated table of results is printed out for each converged solution as it is completed. The user may require more detailed result tables for certain runs and these can be stored on file until the program is terminated. A maximum of five such detailed tables may be stored during one execution. Corresponding data required for plotting purposes is stored simultaneously, if graphical representation is required. (See section 7.435.)

The file in which detailed tables are stored must be assigned before the execution is started as shown in section 7.422.

In the event of an execution-terminating error, or if detailed result tables are not requested by the user, the program automatically deletes PRINTFILE on termination. This is done by adding element FINISH to the runstream through the ERTRAN facility. FINISH contains the following statements:

```
@FREE PRINTFILE.
@DELETE PRINTFILE.
```

If tables are stored but not required, the user must add FINISH to the runstream after termination.

The plotting procedure associated with MULCOM is described in section 7.53.

7.45 Optional enthalpy-calculation subroutine

If the user has enthalpy data available in a form other than those accepted by the standard routine, ENERGY, a separate subroutine may be compiled and used. If the subroutine is named ENTHAL and conforms to the specifications set out below, no alterations to the mainline program, or other subroutines are necessary. Program variables are transferred through the calling list and any data required by the correlations are read through the ERTRAN facility.

The calling program indicates the type of calculation required through counter NEN. The routine is called by the mainline program to determine stream enthalpies throughout the column, or by subroutine QFLASH during the flash calculations.

The enthalpy subroutine must be capable of calculating vapour and liquid enthalpies given composition and temperature data for the streams. It must also compute condenser and reboiler heat duties and an overall feed enthalpy. Enthalpy data returned to the calling program must be in SI units.

An example of a compatible routine is shown below. It may be used directly with MULCOM providing coefficients for the Kobe [55]heat capacity function are available. Alternatively the user may insert other correlations or develop an entirely new routine along the same lines. The heat capacity and latent heat correlations used here are discussed in section 7.234. Both may be extended to multicomponent systems that may be assumed to be ideal [55,58]. Heat of mixing is not considered in this routine, however. The FORTRAN text of ENTHAL is attached, and a description of the calculations performed is shown below.

Line No.*	Program Action
1,2	The calling list is compulsory. Variables are defined below.
3-7	Internal variables dimensioned directly. Calling list variables dimensioned by parameters MC and MP from calling program. Parameter MCC ≥ MC.
10	Conversion factors: CONFAC - Energy units to kW. ENCONV - Enthalpy units to kJ.
11	Read key for UNIVAC system.
12	Read statements bypassed after first calling of ENTHAL.
14-27	Data required by enthalpy correlations read in from data element ENTHDATA. The data shown here are required by the correlations discussed previously (section 7.234). A sample data element is shown in Table 7.43.
(14)	Coefficients - (Equation (7.23)).
(15)	Exponents relating to coefficients BC, CC and DC.

*See FORTRAN text between p.268 and 269.

(16) Latent heats boiling points and critical temperatures.

(17) Molecular weights required if data are not supplied in molar units. (See section 7.234.)

(18) Conversion factors to ensure consistent units in enthalpy calculations.

(19) Base temperature, set at 25°C, adjusted for correlations, temperature units by TCONV.

(22-26) Correction of coefficients.

29,30 Calculation selected by QFLASH.

31-49 DO loops 12 and 13 compute stream enthalpies throughout the column: loop 12 - trays; loop 13 - components.

50-57 Calculation of condenser duty.

58-75 Calculation of feed stream enthalpy.

76 Overall heat balance defines reboiler duty.

77,78 DM and BM required in FLOWNE.
DM - total heat removed per mole of distillate product.
BM - total heat removed per mole of bottoms product.

79,80 Conversion of heat duty units to kW.

The remainder of the routine computes enthalpies for QFLASH for superheated or supercooled feeds, as described above. (NEN = 1)

84-87 TF = specified feed temperature, TD and TB are the dew point and bubble point of the feed respectively.

88-108 DO loops 21 and 22 compute enthalpies of the feed stream as:

(1) Superheated vapour at temperature TF (S1)
(2) Supercooled liquid at temperature TF (S2)
(3) Dew point vapour at temperature TD (S3)
(4) Bubble point liquid at temperature TB (S4)

S1, S2, S3 and S4 are used in QFLASH to determine the feed q-value.

TABLE 7.43

ENTHALPY DATA ELEMENT ENTHDATA

The data shown here correspond to the components shown in Table 7.41.

1	4.750,1.200,0.303,-2.630	Coefficients for Kobe function [55].
2	1.648,4.124,-1.530,1.74	
3	-0.966,7.279,-3.755,7.580	
4	0.945,8.873,-4.38,8.36	
5	1.618,10.85,-5.365,10.10	
6	1.657,13.19,-6.844,13.78	
7	-2,-5,-9	Exponents for coefficients*
8	121.87,-161.6,-82.5	LH_i, T_{BP_i}, T_{C_i}
9	11.87,-88.9,32.1	
10	101.76,-42.1,96.8	
11	92.09,-0.5,153.0	
12	85.38,36.08,-197.2	
13	80.48,68.74.234.8	
14	1.0,16.04	$CPMW_i, LHMW_i$
15	1.0,30.07	
16	1.0,44.09	
17	1.0,58.12	
18	1.0,72.15	
19	1.0,86.17	
20	1.8,1.8,273.	CPCONV,LHCONV,TCONV

* See section 7.23 and Table 7.22

Calling list variables for subroutine ENTHAL

Variable	Type	Description	Dim	Para
AL,AV	real	Moles of liquid and vapour in feed.		
BM,DM	real	Total heat removed per mole of bottoms and distillate product.		
B,D,F	real	Total bottoms, distillate and feed rates		
HD	real	Distillate enthalpy		
HFEED,QF	real	Feed enthalpy - QF* used in printing results. (kJ/kg mole)		
HL,HV	real	Stream enthalpies of liquid and vapour leaving each tray	1	MP
NC	int	Number of components		
NF1,NP1,NP2	int	Feed tray number, total number of trays and condenser number		
NV	int	NV=1 Total condenser NV=2 Partial condenser		
QC,QR	real	Condenser and reboiler heat duties (kW)		
TP	real	Tray temperature (^{o}C)	1	MP
VR	real	Vapour rates leaving each tray	1	MP
X,Y	real	Composition of liquid and vapour streams leaving each tray.	2	MC,MP
XF,XY	real	Composition of each phase in feed stream	1	MC
TB,TD,TF	real	Bubble point, dew point and specified temperature of feed		
NEN	int	Counter selects type of calculation required (from QFLASH or MULCOM).		
S1,S2,S3,S4	real	Phase enthalpies for feed stream (see above)		
NTIME	int	Counter causing subroutine data to be read.		
MC,MP	int	Parameters defining size of arrays from calling program.		

All dimensioned variables with parameter MCC are internal variables. Variables used internally are defined in section 7.234.

*QF may be multiplied by CONFAC to give the feed heat duty in kW.

```
      SUBROUTINE ENTHAL(AL,AV,BM,DM,B,D,F,HD,HFEED,HL,HV,NC,NF1,NP1,NP2,     0001
     1NV,QC,QF,QR,TP,VR,X,XF,Y,YF,TB,TD,TF,NEN,S1,S2,S3,S4,NTIME,MC,MP)     0002
C     CALCULATION OF VAPOUR AND LIQUID ENTHALPY DATA,FEED ENTHALPY,
C     CONDENSER DUTY AND REBOILER DUTY.
C     THIS ROUTINE REQUIRES THE FOLLOWING DATA FOR EACH COMPONENT:
C       (1) COEFFICIENTS FOR A MOLAL HEAT CAPACITY EQN. OF THE FORM:
C         CP=A+B*T+C*T**2+D*T**3  (T IN DEG K,CP IN CAL/GMOL.DEG K)
C     (KOBE K.A.,'THERMOCHEMISTRY FOR THE PETROCHEMICAL INDUSTRY',
C      PETRM. REFNR.,JAN 1949-NOV 1954)
C     SEE ALSO HOUGEN WATSON & RAGATZ,'CHEMICAL PROCESS PRINCIPLES',
C     PART 2,P. XXVI-XXIX.
C       (2) LATENT HEAT AT COMPONENT BOILING PT. (IN CAL/GM.)
C       (3) BOILING PT. OF COMPONENT (IN DEG C)
C       (4) CRITICAL TEMPERATURE OF COMPONENT (IN DEG C)
C       (5) MOLECULAR WT OF COMPONENT
C       (6) CP,LH AND TEMP CONVERSION FACTORS,AND BASE TEMPERATURE.
C     FOR DETAILS REFER TO THE MANUAL,SECTION 4.
C     CONVERSION TO S.I. UNITS (JOULES & WATTS) IS BUILT IN.
      PARAMETER MCC=30                                                      0003
      DIMENSION AC(MCC),BC(MCC),CC(MCC),CPMW(MCC),DC(MCC)                   0004
      DIMENSION HL(MP),HV(MP),LH(MCC),LHMW(MCC)                             0005
      DIMENSION TBP(MCC),TC(MCC),THOLD(4),TP(MP),VR(MP)                     0006
      DIMENSION X(MC,MP),XF(MC),Y(MC,MP),YF(MC)                             0007
      REAL LH,LHCONV,LHMW,LHT                                               0008
    1 FORMAT( )                                                             0009
      DATA CONFAC,ENCONV/0.00027778,2.326/                                  0010
      NCR=8                                                                 0011
      IF(NTIME.EQ.99) GO TO 100                                             0012
      CALL ERTRAN(6,'@ADD,E DISTFILE,ENTHDATA . ')                          0013
      READ(NCR,1) (AC(I),BC(I),CC(I),DC(I),I=1,NC)                          0014
      READ(NCR,1) NBCP,NCCP,NDCP                                            0015
      READ(NCR,1) (LH(I),TBP(I),TC(I),I=1,NC)                               0016
      READ(NCR,1) (CPMW(I),LHMW(I),I=1,NC)                                  0017
      READ(NCR,1) CPCONV,LHCONV,TCONV                                       0018
      TBASE=25.+TCONV                                                       0019
      READ(NCR,1,END=101) IDUM                                              0020
  101 CONTINUE                                                              0021

                                                          SUBROUTINE ENTHAL
```

```
      DO 11 I=1,NC                                                      0022
      BC(I)=BC(I)*10.**NBCP                                             0023
      CC(I)=CC(I)*10.**NCCP                                             0024
      DC(I)=DC(I)*10.**NDCP                                             0025
   11 CONTINUE                                                          0026
  100 CONTINUE                                                          0027
      NTIME=99                                                          0028
      IF(NEN.EQ.1) GO TO 200                                            0029
      IF(NEN.EQ.2) GO TO 103                                            0030
C     CALCULATION OF VAPOUR AND LIQUID STREAM ENTHALPY.
      DO 12 J=1,NP2                                                     0031
      SUMHL=0.0                                                         0032
      SUMHV=0.0                                                         0033
      T=TP(J)+TCONV                                                     0034
      DO 13 I=1,NC                                                      0035
      AFACT=AC(I)*(T-TBASE)                                             0036
      BFACT=BC(I)*(T**2.-TBASE**2.)/2.                                  0037
      CFACT=CC(I)*(T**3.-TBASE**3.)/3.                                  0038
      DFACT=DC(I)*(T**4.-TBASE**4.)/4.                                  0039
      CP=AFACT+BFACT+CFACT+DFACT                                        0040
      H=CP*CPMW(I)*CPCONV                                               0041
      SUMHL=SUMHL+H*X(I,J)                                              0042
      TFACT=(TC(I)-TP(J))/(TC(I)-TBP(I))                                0043
      LHT=LH(I)*(TFACT**0.38)*LHMW(I)*LHCONV                            0044
      SUMHV=SUMHV+(H+LHT)*Y(I,J)                                        0045
   13 CONTINUE                                                          0046
      HL(J)=SUMHL*ENCONV                                                0047
      HV(J)=SUMHV*ENCONV                                                0048
   12 CONTINUE                                                          0049
C     CALCULATION OF CONDENSER DUTY.
      HD=HL(NP2)                                                        0050
      IF(NV.GT.1) GO TO 102                                             0051
C     TOTAL CONDENSER OPERATION.
      QC=VR(NP1)*(HV(NP1)-HL(NP2))                                      0052
      GO TO 103                                                         0053
  102 CONTINUE                                                          0054
```

```
C       PARTIAL CONDENSER OPERATION.
        QC=VR(NP1)*HV(NP1)-(VR(NP1)-D)*HL(NP2)-D*HV(NP2)                    0055
        HD=HV(NP2)                                                         0056
  103   CONTINUE                                                           0057
C       CALCULATION OF FEED STREAM ENTHALPY.
        SUMFL=0.                                                           0058
        SUMFV=0.                                                           0059
        T=TF+TCONV                                                         0060
        DO 14 I=1,NC                                                       0061
        AFACT=AC(I)*(T-TBASE)                                              0062
        BFACT=BC(I)*(T**2.-TBASE**2.)/2.                                   0063
        CFACT=CC(I)*(T**3.-TBASE**3.)/3.                                   0064
        DFACT=DC(I)*(T**4.-TBASE**4.)/4.                                   0065
        CP=AFACT+BFACT+CFACT+DFACT                                         0066
        H=CP*CPMW(I)*CPCONV                                                0067
        SUMFL=SUMFL+H*XF(I)                                                0068
        TFACT=(TC(I)-TF)/(TC(I)-TBP(I))                                    0069
        LHT=LH(I)*(TFACT**0.38)*LHMW(I)*LHCONV                             0070
        SUMFV=SUMFV+(H+LHT)*YF(I)                                          0071
   14   CONTINUE                                                           0072
        HFEED=(AV/F*SUMFV+AL/F*SUMFL)*ENCONV                               0073
        IF(NEN.EQ.2) GO TO 999                                             0074
        QFTOT=F*HFEED                                                      0075
C       CALCULATION OF REBOILER DUTY.
        QR=D*HD+B*HL(1)+QC-QFTOT                                           0076
        DM=HD+QC/D                                                         0077
        BM=HL(1)-QR/B                                                      0078
C       CONVERSION TO S.I. UNITS
        QC=QC*CONFAC                                                       0079
        QR=QR*CONFAC                                                       0080
        QF=HFEED                                                           0081
        GO TO 999                                                          0082

                                                        SUBROUTINE ENTHAL
```

```
  200 CONTINUE                                                          0083
C     CALCULATION OF ENTHALPY FOR Q VALUE DETERMINATION.
      THOLD(1)=TF                                                       0084
      THOLD(2)=TF                                                       0085
      THOLD(3)=TD                                                       0086
      THOLD(4)=TB                                                       0087
      DO 21 J=1,4                                                       0088
      T=THOLD(J)+TCONV                                                  0089
      SL=0.                                                             0090
      SV=0.                                                             0091
      DO 22 I=1,NC                                                      0092
      AFACT=AC(I)*(T-TBASE)                                             0093
      BFACT=BC(I)*(T**2.-TBASE**2.)/2.                                  0094
      CFACT=CC(I)*(T**3.-TBASE**3.)/3.                                  0095
      DFACT=DC(I)*(T**4.-TBASE**4.)/4.                                  0096
      CP=AFACT+BFACT+CFACT+DFACT                                        0097
      H=CP*CPMW(I)*CPCONV                                               0098
      SL=SL+H*XF(I)                                                     0099
      TFACT=(TC(I)-THOLD(J))/(TC(I)-TBP(I))                             0100
      LHT=LH(I)*(TFACT**0.38)*LHMW(I)*LHCONV                            0101
      SV=SV+(H+LHT)*YF(I)                                               0102
   22 CONTINUE                                                          0103
      IF(J.EQ.1) S1=ENCONV*SV                                           0104
      IF(J.EQ.2) S2=ENCONV*SL                                           0105
      IF(J.EQ.3) S3=ENCONV*SV                                           0106
      IF(J.EQ.4) S4=ENCONV*SL                                           0107
   21 CONTINUE                                                          0108
  999 CONTINUE                                                          0109
      RETURN                                                            0110
      END                                                               0111

                                                        SUBROUTINE ENTHAL
```

7.46 Error Messages

Origin	Error Message	Description
MULCOM	DATA NOT ACCEPTABLE ILLEGAL SPECIFICATION ENCOUNTERED.	Error in data input of reply to program question. Program is terminated.
FLOWEQ	ERROR: NEGATIVE LIQUID FLOW IN STRIPPING SECTION CAUSED BY FEED CONDITION Q= ..., LIQUID FLOW= ENTER AN INCREASED REFLUX RATIO TO CONTINUE RUN, OR ENTER 1000. FOR TERMINATION	Mass balance failure. Increase reflux ratio or feed q-value.
FLOWNE	ERROR: NEGATIVE LIQ (OR VAP) FLOW LEAVING TRAY ... POSSIBLE CAUSES OF ERROR... (1)... (2)... (3)...	Enthalpy calculation error probably due to one of the factors suggested by the message or to inconsistent data.
QFLASH	ERROR INFLASH CALCULATION SEARCH FOR FEED TEMPERATURE HAS FAILED VAPOUR FRACTION OF FEED= VALUE SHOULD LIE BETWEEN 0.0 and 1.0. Q= ... FEED TEMP= .. DEG C DEW PT= DEG C BUBBLE PT= DEG C	This error could be caused by inconsistent equilibrium or enthalpy data, or a fault within either the flash or enthalpy routines.
READIN	ONLY TOT OR PAR IS ACCEPTABLE RETYPE NOW.	This is the only data input error searched for. The user is expected to ensure that dimension parameters and data formats are correct.

7.47 Dimension parameters and specification limitations

(a) Dimension parameters.

(1) MC - number of components

(2) MP - number of trays

These are the only two parameters used. They are included in the COMMON block of each subroutine, and in subroutine ENTHAL's calling list. Any internal arrays required by ENTHAL must be dimensioned separately within the routine.

(b) Specification limitations.

(1) Equilibrium and enthalpy data.

Antoine equation is valid for systems at less than 10^3 kPa. The coefficients of other functions used are usually quoted at a given system pressure which is taken at the condenser pressure. The coefficients are assumed to be unaffected by the pressure drop across the trays in the column.

(2) Condenser type.

A total or partial condenser from which a single phase feed is withdrawn, may be considered.

7.5 PLOTTING PROGRAMS

7.51 Application

The plotting program PLOTTER that is included in the distillation package enables the user to construct the following diagrams on a CALCOMP plotter.

1. Multicomponent systems
   (a) Temperature profile.
   (b) Liquid and vapour flowrate profiles.
   (c) Bar diagram showing compositions of distillate product.

(d) Composition profiles through the column.

2. Binary systems.

(a) McCabe-Thiele diagrams for:

(i) Batch distillation operating under constant reflux conditions.

(ii) Batch distillation operating under variable reflux conditions.

(iii) Continuous distillation systems.

(b) Ponchon-Savarit enthalpy-concentration and x-y diagrams for continuous distillation systems.

The program requires access to the standard CALCOMP plotting subroutines and other software. Hardware facilities at the University of Cape Town include a CALCOMP plotter and a Tektronics graphics terminal.

For direct use of the drum-plotter from a standard terminal, the program must be mapped as follows:

(a) @MAP PLOTFILE.PLOTMAP,.PLOTTER

where element PLOTMAP. contains the following statements:

```
IN PLOTFILE.PLOTTER
IN PLOTFILE.PLOTTS
IN PLOTFILE.PLOTFS
IN PLOTFILE.PLOTDS
IN PLOTFILE.PLOTCS
IN PLOTFILE.PLOTBT
IN PLOTFILE.PLOTMT
IN PLOTFILE.AXIS
IN PLOTFILE.MAXIS
LIB CALCOMP*SUBR
```

For execution of PLOTTER from the graphics terminal the following map procedure is used:

```
@MAP PLOTFILE.PLOTNM,.PLOTTER
```

where element PLOTNM contains the following statements:

```
IN PLOTFILE.PLOTTER
IN PLOTFILE.PLOTTS
IN PLOTFILE.PLOTFS
IN PLOTFILE.PLOTDS
IN PLOTFILE.PLOTCS
IN PLOTFILE.PLOTBT
IN PLOTFILE.PLOTMT
IN PLOTFILE.AXIS
IN PLOTFILE.MAXIS
LIB TEKFAST*CALPREV,TEKFAST*TCSAG
```

Plots can be sent to the drum plotter, directly from the graphics terminal, without remapping by entering the instruction.

```
@XQT TEKFAST*CALPREV.TOPLOTTER
```

7.52 Executing PLOTTER in demand mode

The structure of the plotting system at U.C.T. is such that plotting programs are best executed from a terminal. On a standard terminal the CALCOMP map statement ((a) above) automatically queues the job on tape to be plotted as a batch job. The graphics terminal may be used to run the job in demand mode with the plot displayed on the terminal screen (map (b) above). In this way the diagram can be examined before it is sent to the plotter, and discarded if it is unsatisfactory.

PLOTTER is a conversational program, and the following runstream is required to execute it in demand mode. If the user has signed off after executing either BATCH or BINARY, the data file BDPLOT must be assigned with

```
@ASG,AZX BDPLOT.
```

Data files generated by MULCOM are added to the runstream as shown in section 7.53 below.

```
U: @XQT PLOTFILE.PLOTTER.
P: ENTER JOB NO. AS: NO.,DAY,MONTH(INTEGER VALUES).
```

U: 1,2,3 (Run number identifying job)

P: ENTER TYPE OF PLOT REQUIRED.

U: Integer value (code number as tabulated below)

Code	Plot generated	See section
1	Multicomponent plots	7.53
2	McCabe-Thiele diagram batch distillation Constant reflux	7.541
3	McCabe-Thiele diagram batch distillation Variable reflux	7.542
4	McCabe-Thiele diagram continuous distillation	7.551
5	Ponchon-Savarit diagram continuous distillation	7.552

On entering the plot code number the conversational runstream continues as shown below.

7.53 Multicomponent plots

(U: 1)

P: ENTER TOTAL NUMBER OF RUNS TO BE PLOTTED.

U: Integer value (This is the number of runs completed within a single execution of MULCOM as described in section 7.4. The maximum value is 5.)

P: ENTER NUMBER OF RUN FOR WHICH FULL PLOT IS REQUIRED.

U: Integer value (The program will superimpose the temperature profiles for all the runs on one diagram, but the remaining diagrams will show data for a single run only.)

P: ENTER NUMBER OF TRAYS IN COLUMN.

U: Integer value (exclude reboiler and condenser)

P: ENTER NUMBER OF COMPONENTS IN FEED.

U: Integer value

P: ENTER DATA FILE NAMES.

U: File names used in execution of MULCOM, in correct order. e.g. from section 7.435.
DFILE1,DFILE2,DFILE3,DFILE4
The files are then assigned automatically through ERTRAN. The final output from the computer is generated by EXEC 8.

i.e. P: PLOT ENTERED IN QUEUE.
The user may then sign off with

U: @FIN

The plot is then collected at the Computer Centre (turn-around time ∿ 24 hours).

7.54 Binary batch distillation

7.541 Constant reflux operation:

(U: 2)

No further instructions are required. On completion of the plot the computer responds

P: PLOT ENTERED IN QUEUE.

U: @FIN.

7.542 Variable reflux operation:

(U: 3)

As for constant reflux operation above.

7.55 Binary continuous distillation

7.551 McCabe-Thiele disign

(U: 4)

As for section 7.541.

7.552 Ponchon-Savarit design:

(U: 5)

P: ENTER UNITS OF ENTHALPY AXIS IN ENTH-CONC DIAGRAM. NORMALLY THESE ARE DEL POINT VALUES ROUNDED CONVENIENTLY. ENTER UPPER LIMIT FIRST. (REAL MODE, FREE FORMAT).

U: Δ_D, Δ_B enthalpy values

These values are printed on the result table in E12.6 format as

P: ENTHALPY DIAGRAM DEL POINTS:
DEL D = ... KJ/KGMOLE
DEL B = ... KJ/KGMOLE

Consider DELD = .341099+05 and DELB = -.213842+05. The user should enter the following values in the plotting runstream

U: 35000.,-22000.

On completion of the plot the computer responds

P: PLOT ENTERED IN QUEUE.

U: @FIN.

For batch mode execution of PLOTTER, see section 7.63.

7.56 Diagram titles

All titles written on diagrams constructed by PLOTTER are contained in separate elements which are accessed through the ERTRAN facility as required. This structure has been adopted to enable users to alter the titles without having to edit the main program. The format of the elements, as shown below, must be rigidly adhered to, however. Each element contains certain alphanumeric labels and integer values corresponding to the number of characters in each label. The elements are stored in PLOTFILE.

(a) Multicomponent diagrams:

```
  TITLE ELEMENT: HEADING
TOTAL NUMBER OF RUNS PLOTTED:
FULL PLOT PRODUCED FOR RUN NO:
JOB NUMBER:
29,30,11
  TITLE ELEMENT: TITLES
TEMPERATURE PROFILE
19
VAPOUR AND LIQUID FLOWRATES
27
DISTILLATE COMPOSITIONS
23
COMPOSITION PROFILE
19
THIELE-GEDDES PROGRAM
21
MULTICOMPONENT DISTILALTION
27
0.1,9.5,0.1,9.0,0.1,8,5
0.1,9.5,0.1,9.0,0.1,8.5
0.1,6.5,0.1,6.0,0.1,5.5
0.1,6.5,0.1,6.0,0.1,5.5
  TITLE ELEMENT: AXISTITLE
TRAY NUMBER
11
TEMPERATURE DEG C
17
TRAY NUMBER
11
LIQ. AND VAP. RATES KGMOL/HR
28
COMPONENT NO.
13
MOLE FRACTION
13
TRAY NUMBER
11
MOLE FRACTION
13
8.,8.,5.,5.
```

(b) Binary system diagrams:

```
  TITLE ELEMENT: BBVTTL
BINARY BATCH DISTILLATION
VARIABLE REFLUX RATIO
X-Y DIAGRAM
JOB NUMBER
25,21,11,11
```

```
  TITLE ELEMENT: BBCTTL
BINARY BATCH DISTILLATION
CONSTANT REFLUX RATIO
X-Y DIAGRAM
JOB NUMBER:
25,21,11,11

  TITLE ELEMENT: AXISMT
LIQUID COMP.,MOLE FRACTION
VAPOUR COMP.,MOLE FRACTION
26,26

  TITLE ELEMENT: AXISPS
X,Y ,Liq and vap mole frac
ENTHALPY,KJ/KGMOLE SOLUTION
26,27

  TITLE ELEMENT: BCTITL
BINARY CONTINUOUS DISTILLATION
MCCABE-THIELE DIAGRAM
JOB NUMBER:
30,21,11

  TITLE ELEMENT: PSTITL
BINARY CONTINUOUS DISTILLATION
PONCHON-SAVARIT DESIGN
ENTHALPY-CONCENTRATION DIAGRAM
JOB NUMBER:
30,22,30,11
BINARY CONTINUOUS DISTILLATION
PONCHON-SAVARIT DESIGN
X-Y DIAGRAM
30,22,11
```

7.6 SUMMARY OF GENERAL OPERATING PROCEDURE FOR ALL PROGRAMS

7.61 Before signing on at a terminal or compiling a batch job card deck:

(1) Determine whether the program is able to solve the problem and compile a table of all data required.

(2) Check that the conversion factors included in program will convert the units in which data is supplied to those required in the calculations. These conversion factors are defined in DATA statements in the mainline routines and are described in the relevant sections of this manual.

(3) Check that the dimensioning parameters are large enough to accommodate the system considered. These are

defined in PARAMETER statements in the COMMON block of each program.

(4) Set up data elements containing equilibrium, enthalpy or other bulky data tables required during the run. Physical property data required by PACKED must be inserted in specific elements as shown in section 7.34.

7.62 After signing on at a terminal in demand mode:

(1) Assign the data files required by the plotting program, if this facility is to be used. The necessary instructions are given in each section of the manual.

(2) Initiate the run through an @XQT instruction.

(3) Follow the program's conversational instruction carefully.

(4) Execute plotting program only after analysing the program results.

7.63 When compiling a card deck for batch mode execution:

(1) Ensure that the following instructions are included and that the data cards correspond exactly with the program's instructions. The conversational data input facility should be suppressed.

```
@RUN card
@ASG,AX DISTFILE.
@XQT DISTFILE.program name
Data cards, or @ADD DISTFILE. data element
@FIN
```

(2) If the plotting facility is executied in batch mode the following instructions are required. The program must be mapped as shown in section 7.51(a).

@RUN card
@ASG,AX BDPLOT (required for binary systems only)
@XQT PLOTFILE.PLOTTER
Data cards for required plotting procedure (see section 7.5).

Note: The computer solution to a problem must always be checked for errors before submitting a plotting job, to prevent wastage of plotting time.

7.64 General rules for data input

(1) All numerical data must be supplied in free format.

(2) Names of components must not exceed 30 characters (5A6 format).

(3) All questions must be answered YES or NO unless otherwise indicated.

(4) Column pressure may be specified in any of the following units. (1) mmHg, (2) psig, (3) psia, (d) kPa, (e) atm.

(5) Flowrates must be specified in kg moles/hr and temperature data in ^{o}C unless otherwise indicated.

(6) All single composition values in binary systems refer to the light component. In multicomponent systems component 1 is the lightest and component NC is the heaviest.

(7) Tray numbering convention: Reboiler = 0; Column Trays = 1-N; Condenser = N+1; Feed tray position = Tray NF from bottom of column.

(8) Data tables based on composition must list data from x=0 to x=1.

For information concerning the program algorithms, FORTRAN text, or other computer software, the user is referred to the original thesis.

CHAPTER 8

PROGRAM ANALYSIS AND APPLICATIONS

8.1 COMPUTER PROGRAMS AS TEACHING AIDS

A comprehensive set of computer programs designed specifically for undergraduate courses in chemical engineering has been published by CACHE. These programs have been developed by a number of workers, however, and do not form an integrated package with interchangeable subroutines. Many of them are examples of programs that must be written by students in tutorial assignments and thus do not cover their specific subjects in much detail. Also, certain topics, such as McCabe-Thiele designs, are considered in several programs whereas others, such as Ponchon-Savarit designs, have been ignored. It was thus decided to develop a fully integrated set of programs covering all the important aspects of distillation considered in the undergraduate course in mass transfer operations at U.C.T.

Students are not required to write programs in this course, but rather to use the programs available to study the behaviour of distillation systems in detail. The programs were therefore to be economical in computer time and core storage requirements, as well as being simple to execute and able to handle a wide variety of problems. As a teaching aid, the program should be able to guide a user by means of descriptive comments and error messages. The previous discussion of each program has shown how error messages and conversational data input facilities have been included to assist users who have no more than a basic knowledge of the principles involved. The techniques used to reduce the computer time and core requirements of each program are discussed below. Conventions adopted to facilitate future alterations to the program texts are also described.

8.2 PROGRAMMING CONVENTIONS

In order to increase the versatility of the programs certain options are included that would not normally be required for undergraduate tutorials. Options such as providing a multicomponent enthalpy routine or data for different packing materials require the user to alter the FORTRAN text of the standard routines. To assist users in altering the programs or adapting them for other applications, the following conventions have been used in all mainlines and subroutines.

(1) READ and WRITE keys:

Read and Write keys are defined in DATA statements at the beginning of each mainline program. They are included in the COMMON blocks and, where necessary, in the calling lists of subroutines without common variables. Names and definitions of these keys, that are common to all programs unless otherwise stated, are listed below.

Name	Integer Used	Purpose
NCR	8	General READ key on UCT's UNIVAC 1106
NTP	5	General WRITE key on UCT's UNIVAC 1106
NG1	11**,12*	Write to file for plotting use
NG2	13*	" " " " " "
NG3	14*	" " " " " "
NG4	15*	" " " " " "
NLP	16*	Write to file for detailed printout of results

* Used in multicomponent program MULCOM only.

** Used in binary programs BATCH and BINARY.

Data files are not required by the program PACKED.

(2) Statement numbers - general use:

Range	Application
1-9	READ FORMAT statements
1000-9999	WRITE FORMAT statements
10-99	DO LOOPS
100-999	GO TO and IF statement addresses.

(3) Statement numbers - specific use:

The statement numbers indicate the section of program in which they fall and, in general, the type of operation being performed according to the following convention. This has been adopted to facilitate tracing errors or altering the programs. The convention applies throughout the package.

Range	Application
50-59,500-599,5000-5999	Input of data or selection of options.
80-89,800-899,8000-8999	Compilation and writing to file of data for plotting purposes.
90-99,900-999,9000-9999	Error messages and program termination procedures.

All other numbers are used in blocks for each stage of the particular calculation procedure, e.g. in BATCH.

10-19,100-199,1000-1999	Variable reflux operation case.
20-29,200-299,2000-2999	Constant reflux operation case.
40-49,400-499,4000-4999	Printing results.

(4) Choice of variable names and other nomenclature:

The nomenclature list defines most of the variables, appearing in the programs. In general, the same variable names are used throughout the package although in some cases it was convenient to use a binary program variable name for a different purpose in the multicomponent program.

(a) DO loop counters are chosen to correspond to text nomenclature.

I - component number

J - tray number in loops covering the whole column

N - tray number in loops covering the rectifying section only

M - tray number in loops covering the stripping section only

K - general counter used for loops not categorized as above.

(b) Variable names are chosen to describe their application.

Alphanumeric character	Association when used as first or second character in a name
B	Bottoms product
D	Distillate product
F	Feed stream
G	Vapour or steam entering bottom of column
H	Enthalpy
L	Liquid stream (real variables)
P	Pressure
Q	Feed q-value or heat duty
R	Rectifying section
S	Stripping section
T	Temperature
V	Vapour
X	Liquid composition
Y	Vapour composition
Z	Feed composition

In addition variable names similar to standard textbook nomenclature are used where possible.

(5) Tray numbering:

The standard tray numbering convention designates the reboiler as stage 0 and the condenser as stage N+1 [11]. It is computationally convenient to operate on a base number of 1, rather than 0, therefore the internal tray numbering relates to the conventional system as follows:

Conventional system		Internal system
Reboiler	0	1
Bottom tray	1	2
Feed tray	N_F	NF1 = NF+1
Top tray	N	NP1 = N+1
Condenser	N+1	NP2 = N+2

Thus the figures supplied as the feed tray position and the top tray number are numerically equal to the tray below the feed and the tray below the top tray respectively. The user is not concerned with this convention when operating the program. Input and output tables conform to the standard convention.

8.3 COMPUTER CORE REQUIREMENTS

As computer systems with large core storage capacities are not always available for teaching purposes, it is, in general, necessary to keep core requirements of teaching programs to a minimum. Although the UNIVAC 1106 machine at U.C.T. is a teaching computer with a large core area (265 K), the annually increasing load imposes limits on the size of program that may be run in demand mode. For this reason, the core requirement of each program has been limited as follows:

(1) Number of source cards reduced by use of subroutines for repetitive calculations.

(2) Use of dimensioning parameters that can be altered by the user to minimise the core requirement for a given

problem. Separate parameters are used for different blocks of variables so that all arrays need not necessarily be expanded for a particular application.

(3) Use of a single array for more than one purpose in the calculations. This has not been extensively applied, however, because most of the arrays in the programs are required throughout the calculations. Arrays have not been equivalenced for the same reason.

(4) Use of the segmented mapping facility on the UNIVAC system at U.C.T. This facility causes subroutines associated with a program to be transferred to core only when they are required. Core savings of 10% or more are realized in this way.

The core requirements of each program compare favourably with those of CACHE programs, which are assumed to be suitable for teaching purposes. CACHE programs with similar capabilities and using similar calculation procedures have been chosen for comparison.

(a) Program BATCH

CACHE program VI-5 is a batch distillation program assuming negligible holdup and McCabe-Thiele conditions. It considers both constant and variable reflux operations in columns with a known number of trays. Mathematical techniques used include linear interpolation and integration by quadrature. The program requires 16 K of core on an IBM 360 machine.

BATCH requires 10 K of core in order to handle columns with up to 25 trays. This number has proved to be sufficient for all tutorial problems found in the literature. Segmented mapping did not reduce the core requirements appreciably in this case (i.e. < 1 K).

Additional facilities offered in BATCH include

(1) Partial or total condenser operation.

(2) Specification of average distillate composition for constant reflux operation.

(3) Three methods for providing equilibrium data. (Linear interpolation of equilibrium data is the only technique offered in the CACHE program.)

(4) Writing data to file for plotting of x,y diagram for the system.

(b) Program BINARY

Several programs dealing with McCabe-Thiele designs have been published by CACHE. Program VI-6 is similar to the McCabe-Thiele facility in BINARY in that it calculates the number of trays required to achieve a given separation, allowing for a Murphree tray efficiency factor. In addition, the effect of liquid entrainment may be studied and a Newton-Raphson flash calculation is included. Equilibrium data are determined using equilibrium constants expressed as a function of temperature. This program requires 30 K of core on an IBM 360.

BINARY includes both a McCabe-Thiele and a Ponchon-Savarit design procedure and requires 18 K of core with a standard map, and 15 K when segmented. This appears to be a reasonable requirement by comparison with the CACHE program. Additional facilities offered by BINARY include:

(1) Ponchon-Savarit design procedure.
(2) Open steam replacement of a partial reboiler in McCabe-Thiele designs.
(3) Three methods for providing equilibrium data.
(4) Operating diagram data for plotting purposes.
(5) Total or partial condenser.

Other CACHE programs require less core than VI-6, but none are as comprehensive. Ponchon-Savarit design is not considered by any of the CACHE programs.

(c) Program PACKED

CACHE do not consider distillation in packed columns. A program for designing packed absorption towers, using mass transfer coefficient correlations, has been published, and this is used for comparative purposes. The program (VI-2) is similar to PACKED in its approach to the design of packed columns. Column diameter and packed height are obtained, based on the characteristics of the packing and the physical properties of the fluids. The mass transfer correlations used, however, restrict the program to systems of light hydrocarbons and water, and a packing of Raschig rings. The program requires 40 K and a CDC 6600 machine.

PACKED requires ∿ 27 K of core with a standard map and 23 K when segmented. The correlations and data included enable the program to consider a wide variety of systems, and packings of either Raschig rings or Berl saddles. PACKED also allows the user to specify either the column diameter or an allowable pressure drop and then computes the corresponding unknown.

Like PACKED, CACHE program VI-2 was written as a teaching aid to be used by students. It is, however, considerably larger and is unsuitable for distillation column design.

(d) Program MULCOM

A number of programs dealing with multicomponent systems have been published by CACHE. Program VI-13 uses the Thiele-Geddes calculation procedure combined with Holland's θ method of convergence to rate distillation columns of known configuration. The program will consider ideal systems only and computes equilibrium data using Antoine's equation and Raoult's law. The program is thus similar to MULCOM in many

respects. It does not, however, consider enthalpy effects, tray efficiency or two phase feeds. The CACHE program requires 43 K on an IBM 360 machine whereas MULCOM requires 21 K with a standard map and 17 K when segmented. These requirements apply when MULCOM is dimensioned to accommodate ten components and 25 trays which are sufficient for most tutorial problems. If the dimension parameters are increased to thirty components and fifty trays, 32 K of core is required. For teaching purposes, however, this would be excessive.

A program capable of handling complex columns has been published by CACHE. The program (VI-16) uses the Wang Henke [25] algorithm, which is a tridiagonal matrix technique combined with Muller's iterative convergence procedure. This program considers enthalpy effects, column heat losses and an arbitrary number of single phase feed and side streams. A disadvantage of the program is that equilibrium and enthalpy data are computed using polynomials in temperature for which coefficients must be provided. Data for the sample problem shown in the CACHE manual was extracted from the original paper of Amundson and Pontinen [4]. The determination of coefficients for arbitrary tutorial problems is usually time consuming and for this reason optional procedures are offered in MULCOM. Tray efficiency considerations and flash calculations are not included in the CACHE program either. The program requires 29 K of core on a Univac 1108, which is 6 K smaller than MULCOM for systems of the same size. It is, however, more difficult to use for general teaching purposes because of the limited data options.

It can therefore be said that the programs developed in this study offer more facilities and require no more core than comparable CACHE programs.

8.4 CPU TIME REQUIREMENTS

The question of run times is discussed far more extensively in the literature than that of core requirements. In industry, where large computers are usually available, a major cost is the CPU time required to run a program. In a teaching environment, where the number of jobs run is often very high, a minimal use of CPU time is imperative. Direct comparison of run times of programs executed on different machines is not valid because some CPU's are faster than others, and it is not always clear whether a quoted run time refers to CPU time or total time. The mathematical techniques used in this study were chosen:

(1) to produce stable algorithms,
(2) to correspond to material presented in the course, and
(3) to reduce the CPU time required.

The first two points have been discussed previously but each identifiable technique will now be discussed briefly with respect to all three considerations.

(a) Newton's method of convergence.

Also known as Newton's method of tangents or, when applied to multivariable systems, as the Newton-Raphson procedure, this technqiue is known to be unstable if the starting value is not a reasonable estimate of the end result. This potential instability means that the procedure should be applied only after a careful study of the behaviour of the function in the region of the desired root. If the function is not well behaved, a poor initial estimate often leads to spurious results or divergence. In this study the method is applied to bubble point and dew point calculations, and in Holland's θ method. In these applications the functions are wellbehaved and, in the case of temperature calculations

pure component boiling points are satisfactory starting values. For the θ method (see Chapter 5) any positive value close to zero will ensure stable convergence. It is generally acknowledged that this method is the fastest available providing the stability is checked. Newton's method is introduced in the early part of the course when bubble point calculations are discussed. It is also introduced as a computer technique in an elementary course on computer programming.

(b) Direct iteration: (Subroutine SEARCH).

Direct iteration guarantees convergence providing that changes are made in the correct direction. Changes are linear, however, and thus the method is comparatively slow. It has been applied in this study where stability is of paramount importance, and where the behaviour of the function is not always known. Iteration run times of less than a few milliseconds for BATCH and BINARY, where it is used extensively, indicate that it is sufficiently fast for these applications. The method has been applied for similar problems in several CACHE programs. Trial and error methods are the basis of most chemical engineering design procedures, and direct iteration is therefore discussed early in the curriculum.

(c) Linear Interpolation.

This is another basic procedure that is used chiefly in the binary programs. For well-behaved functions such as ideal system equilibrium curves, the accuracy is satisfactory providing the interpolation interval does not exceed 0,05 mole fraction. The alternative of fitting a polynomial to the data is far more time consuming, and the improved accuracy does not justify the increased CPU time requirement. The use of subroutine FDATA is discussed below. A run time of 0,6 sec (CPU) for solving a McCabe-Thiele complex column problem (section 8.5.2) indicates that the procedure is satisfactory. It is used in BATCH and BINARY, as in certain CACHE programs,

(VI-5, VI-8), for stepping off stages on the x-y diagram, as well as other applications discussed previously.

(d) Fitting of polynomials: (Subroutine FDATA).

A least squares procedure is used to fit a polynomial to a specified flooding and loading correlation curve. Linear interpolation cannot be applied satisfactorily to data extracted from a log-log plot. In addition the least squares facility, which tests up to 10th order polynomials, enables users to consider more specific pressure drop correlations if data are available. Although slower than linear interpolation, it is able to handle complex functions with greater accuracy.

(e) Holland's θ method.

This has been discussed extensively in Chapter 5. It is acknowledged as one of the fastest convergence procedures for multicomponent distillation systems [34], and the Thiele-Geddes-θ combination has proved to be stable for a wide variety of simple distillation systems. In this application the inclusion of two convergence criteria reduces the CPU time required for specific problems, where one criterion is met before the other. Limiting the maximum value of θ, and preventing negative values, also improved the stability.

(f) Forcing procedures.

Three forcing procedures were found to improve the rate of convergence in MULCOM, and thus reduce the CPU time required:

(1) Limitation of absolute change in flowrate from one iteration to the next (subroutine FLOWNE) as proposed by Lyster [33].

(2) Limitation of absolute change in temperature profile from one iteration to the next (subroutines BUBCAL and DEWCAL) as proposed by Lyster [33].

(3) Averaging of bubble point and dew point on each tray to determine new temperature profile. Previous workers [18, 21,33] have suggested the use of either dew point or bubble point calculations through the column to establish new temperature profiles. Holland also found that some systems converged faster if dew point calculations were used in the rectifying section and bubble point calculations were used in the stripping section of the column. In this study it has been found, in general, that the use of both dew point and bubble point calculations throughout the column, with subsequent averaging of the two values on each tray, reduces the total number of iterations required for convergence by up to 30%. This procedure proved to be stable for all problems attempted.

(g) Simpson's Rule numerical integration.

This method is introduced in an elementary course on computer programming and is thus familiar to undergraduates. It is acknowledged as an accurate numerical method although the absolute accuracy depends largely on the integration interval. From a comparison of run times of the CACHE program VI-5, which integrates the batch distillation equations by quadrature, and BATCH, it appears that the Simpson's Rule procedure is faster. This conclusion is based on the similarity of the other calculation procedures (e.g. stepping off stages using linear interpolation). Comparative run times are

	CACHE VI-5 (IBM 360-67)	BATCH (UNIVAC 1106)
Constant Reflux:	5-10 mins	8-10 secs.
Variable Reflux	1-2 mins	6-8 sec.

It should be noted, however, that CACHE run times are quoted as "typical" run times, and it is not clear whether they are CPU or Total times.

Typical CPU run times for each program developed in this study have been listed in the summary sheet (section 6) of each chapter. These times are all within the limits of 20 secs and 30 K core set by the Computer Centre U.C.T. for demand mode jobs at the time of writing.

8.5 PROGRAM APPLICATIONS

In this section solutions generated by each program are discussed and compared where possible with data presented in the literature.

8.51 Batch distillation - program BATCH

The program solutions to the following problems demonstrate the operation and capabilities of BATCH.

Problem 8.51(a) (Variable reflux operation)

(Data from Treybal [52] p.335.)

Fifty kgmoles of a solution of CS_2 and CCl_4 is charged to a batch fractionator kettle. The composition of the charge is 0,4 mole fraction CS_2 and a distillate containing 0,95 mole fraction CS_2 is required. The column contains 6 ideal trays and is limited to a maximum boilup rate of 10 kgmoles/hr.

Assuming a total condenser and a column pressure of 101,3 kPa, determine the time required to achieve a final residue concentration of 0,08 mole fraction CS_2.

TABLE 8.511

SOLUTION TO PROBLEM 8.51(a)

```
BATCH DISTILLATION OF A BINARY SOLUTION.
VARIABLE REFLUX OPERATION.

FEED COMPONENTS: (1) CARBON DISULPHIDE
                 (2) CARBON TETRACHLORIDE

INITIAL REFLUX RATIO: 1.487      (1.75)
FINAL REFLUX RATIO:   9.800      (10.88)
```

VAPOUR BOIL-UP RATE: 10.00 KGMOLE/HR
BATCH CYCLE TIME: 7.441 HRS (8.06)

TOTAL VAPOUR BOILED UP DURING CYCLE: 74.412 KGMOLE (80.6)

RESIDUE: 31.609 KGMOLE (31.6)
RESIDUE COMPOSITION: .0800 (0.08)

TOTAL DISTILLATE WITHDRAWN: 18.391 KGMOLE (18.4)

DISTILLATE COMPOSITION: .9500 MOLE FRAC (0.95)

The problem is solved using Antoine constants to generate equilibrium data. Table 8.511 shows a cycle time of 7,44 Hrs and Figure 8.511 is a McCabe-Thiele diagram, constructed by PLOTTER, showing the initial and final operating conditions.

The results of Treybal [52] agree well as shown by the values in parenthesis in Table 8.511. Discrepancies are probably due to differences in equilibrium data. The system is slightly easier to separate if Antoine constants correctly describe the equilibrium.

Solving the same problem assuming a constant relative volatility of 3,5, yields a cycle time of 5,01 Hrs. This is expected because the shape of the equilibrium curve changes to increase the distance between it and the operating line. Thus the separation is easier and lower reflux ratios may be used (initial R = 0,89; final R = 5,06).

Problem 8.51(b) (Constant reflux operation)

One hundred kgmoles of CCl_4 and toluene (composition 0,3 mole fraction CCl_4) is to be distilled at 1 atm, at a constant reflux ratio of 5,0. The column contains 5 theoretical trays and is limited to a vapour boilup rate of 8 kgmoles/hr. A total condenser is used.

1) Determine the quantity of distillate product that may be withdrawn if the average concentration required is 0,90 mole fraction CCl_4.

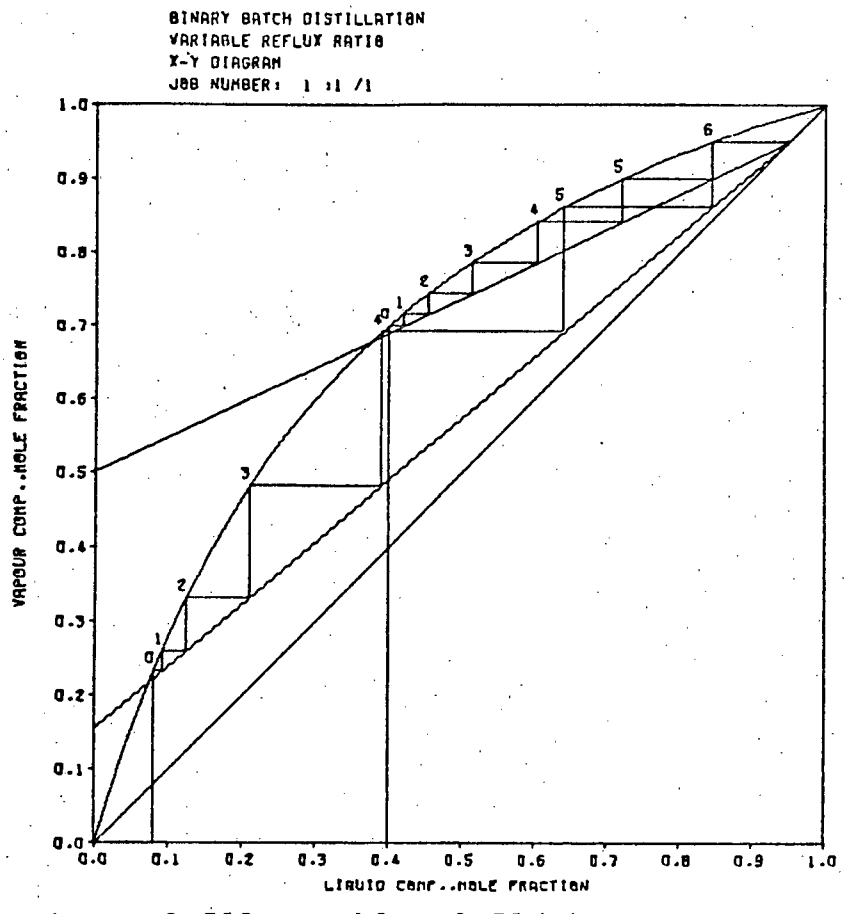


Figure 8.511 Problem 8.51(a) - x,y diagram.

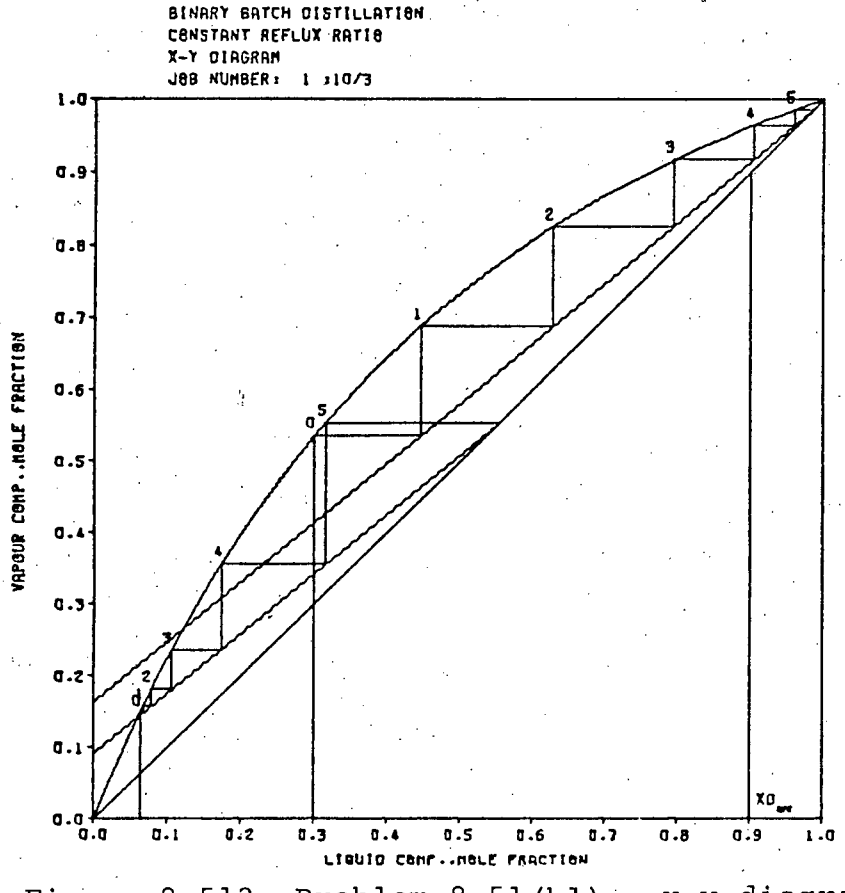


Figure 8.512 Problem 8.51(b1) - x,y diagram.

2) Determine the average distillate composition if the final residue composition required is 0,05 mole fraction CCl_4.

TABLE 8.512

SOLUTION TO PROBLEM 8.51(b2)

BATCH DISTILLATION OF A BINARY SOLUTION.
CONSTANT REFLUX OPERATION.

FEED COMPONENTS: (1) CARBON TETRACHLORIDE
(2) TOLUENE

REFLUX RATIO: 5.000

VAPOUR BOIL-UP RATE: 10.00 KGMOLE/HR
BATCH CYCLE TIME: 18.253 HRS

TOTAL VAPOUR BOILED UP DURING CYCLE: 182.534 KGMOLE

RESIDUE: 69.578 KGMOLE
RESIDUE COMPOSITION: .0500

TOTAL DISTILLATE WITHDRAWN: 30.422 KGMOLE

INITIAL DISTILLATE COMPOSITION: .9862 MOLE FRAC
FINAL DISTILLATE COMPOSITION: .4517
AVERAGE DISTILLATE COMPOSITION: .8718

Table 8.512 shows an average distillate composition of $x_{D_{AVE}}$ = 0,8718, with a cycle time of 18,2 Hrs. Repeating the problem for a specified $x_{D_{AVE}}$ = 0,9, the cycle time is reduced to 16 Hrs whereas the residue composition increased (0,06 mole fraction) and the distillate product yield is reduced to 28 kgmoles. Figure 8.512 shows a McCabe-Thiele diagram for the system with $x_{D_{AVE}}$ specified. Initial and final conditions are shown.

Figures 8.511 and 8.512 are code 3 and 2 plots respectively (see User Manual, section 7.54). In addition to the results shown in Tables 8.511 and 8.512, BATCH prints out a table of either reflux ratio (variable reflux operations) or distillate

composition (constant reflux operations) vs. residue composition. These data may then be used to check the integration manually.

8.52 Binary distillation in tray columns - program BINARY

The graphical design methods of McCabe and Thiele, and Ponchon and Savarit are discussed in detail in the undergraduate course. Program BINARY may be used to demonstrate a number of features associated with these design procedures.

8.521 McCabe-Thiele Problems: Figures 8.521 and 8.522 show complex columns operated under McCabe-Thiele conditions of equimolal overflow. Figure 8.521 shows a system with two feed streams under open steam operation. A Murphree tray efficiency factor of 0,75 is assumed and a total of 7,8 trays are required. If perfect stages are assumed this number is reduced to 5,7 and if a partial reboiler is used only 3,9 ideal trays are required. Figure 8.522 shows a column with a side stream withdrawal and a partial reboiler. Ideal trays are assumed in this case. The result tables produced by BINARY list the flowrates in each section of the column, and the trays on which feeds are entered or from which side streams are withdrawn.

Figures 8.523 and 8.524 demonstrate the use of Antoine constants and equilibrium data respectively for the same system (see Problem 8.521 below). The shape of the equilibrium curve affects the feed tray position, and the total number of trays required.

Problem 8.521

(System described by Figure 8.524.)

A solution of methanol and water containing 0,38 mole fraction methanol is distilled at a rate of 100 kgmoles/hr in a column fitted with a partial reboiler and a partial condenser. A two phase distillate product is withdrawn with

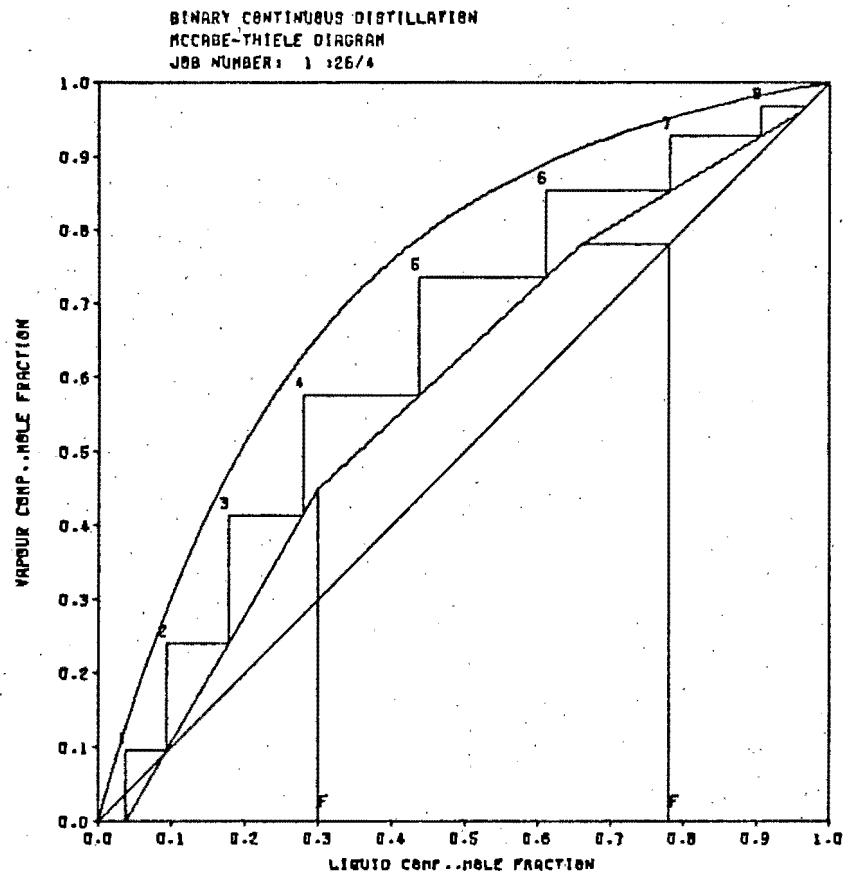


Figure 8.521 Complex column - open steam operation.

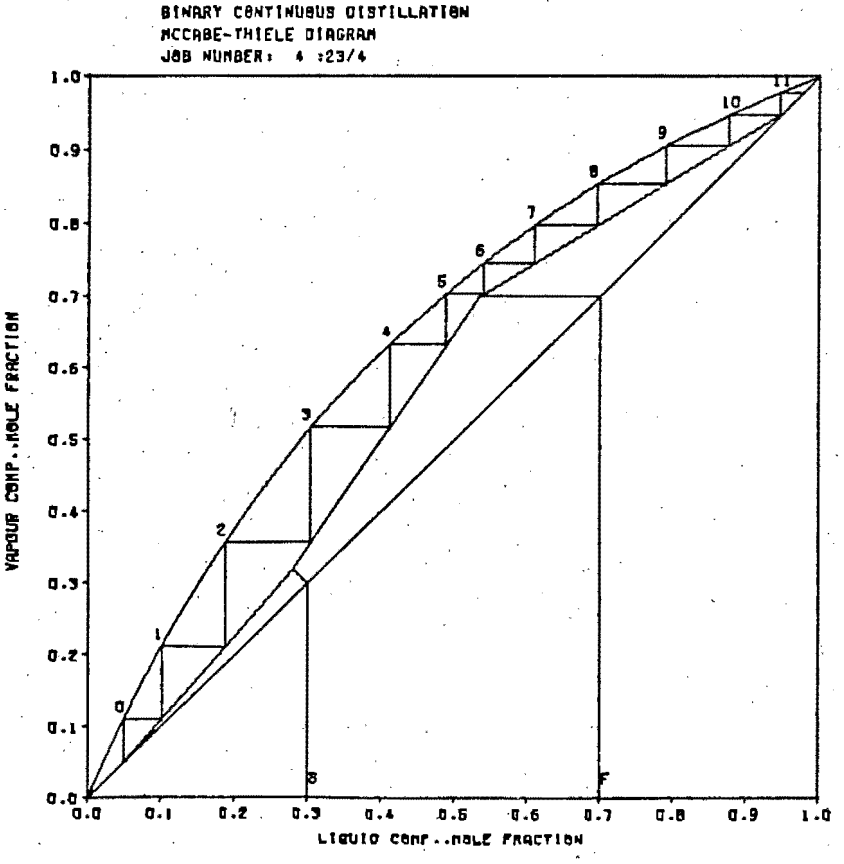


Figure 8.522 Complex column - partial reboiler used.

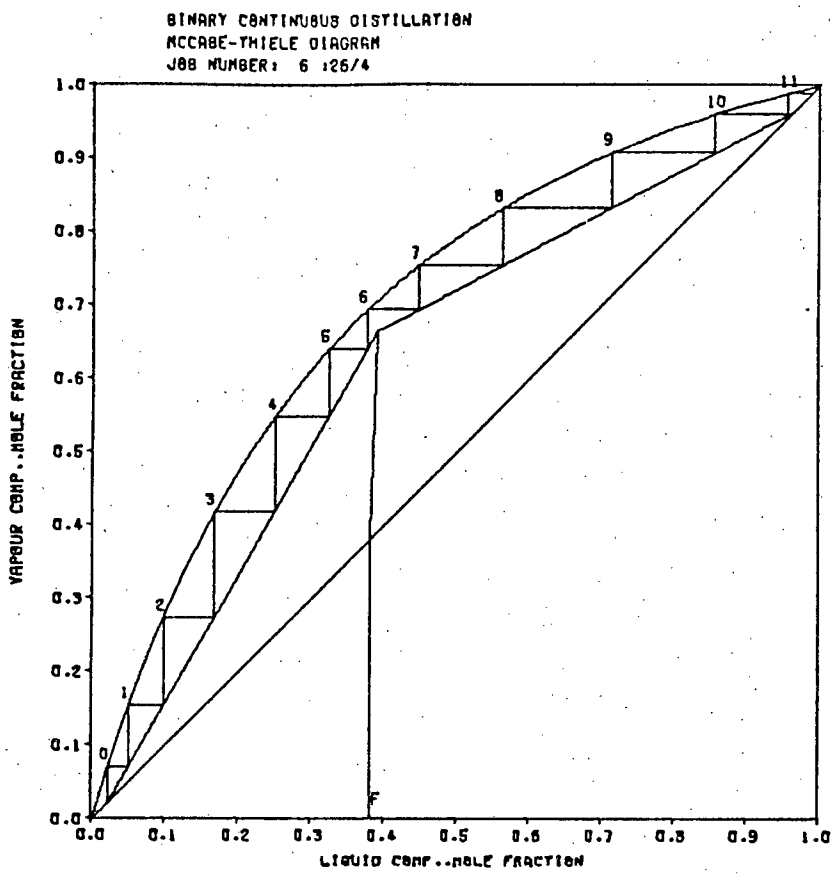


Figure 8.523 Problem 8.521 - Antoine constants.

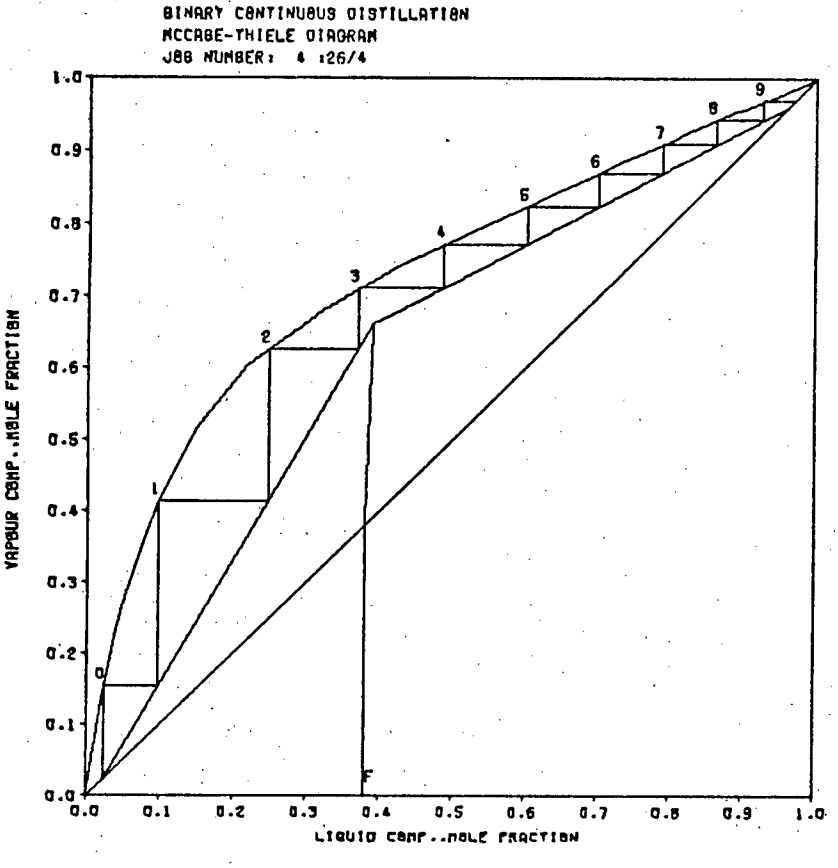


Figure 8.524 Problem 8.521 - x,y equilibrium data.

a liquid phase composition of 0,95 mole fraction. The total distillate withdrawal rate is 38,0 kgmoles/hr, of which 60% is liquid.

Determine the minimum number of ideal trays required in the column, and the feed tray position, if the feed is a liquid just below its bubble point (q = 1,04) and the reflux ratio is set at 1,1. Assume that the column operates at atmospheric pressure.

TABLE 8.521

SOLUTION TO PROBLEM 8.521

MCCABE-THIELE DESIGN OF A BINARY DISTILLATION SYSTEM.

FEED COMPONENTS: (1) METHANOL
(2) WATER

NO. OF FEED STREAMS: 1
NO. OF SIDE STREAMS: 0

NO. OF THEORETICAL STAGES REQUIRED: 8.4

MURPHREE EFFICIENCY FACTOR USED: 1.00

FEED AND SIDE STREAM SPECIFICATIONS:

STREAM NO	TYPE	TRAY NO	Q VALUE	FLOWRATE KGMOLE/HR	COMPOSITION MOL FRAC
1	FEED	3	1.04	100.00	.3800

PARTIAL CONDENSER OPERATION
CONDENSER PRESSURE: 101.32 KPA

REFLUX RATIO 1.1000

DISTILLATE WITHDRAWAL RATE: 38.00 KGMOLE/HR
DISTILLATE COMPOSITION: .9616

BOTTOMS WITHDRAWAL RATE: 62.00 KGMOLE/HR
BOTTOMS COMPOSITION: .0235

VAPOUR AND LIQUID RATES THROUGH COLUMN:

		VAPOUR	LIQUID	(KGMOLE/HR)
SECTION 1	REBOILER - FEED (1)	83.80	145.80	
SECTION 2	FEED (1) - CONDENSER	79.80	41.80	

The distillate composition shown is the overall value calculated by the program. In addition to this table, a table of data describing the McCabe-Thiele operating diagram may be requested. These data may be used to construct the diagram manually if a plotting facility is not available.

The effect of feed condition, condenser type, reflux ratio and tray efficiency may be investigated using BINARY. Figure 8.525 shows the system described in problem 8.521 assuming a Murphree tray efficiency factor of 0,8. Note that the reboiler (Stage 0) is assumed to be an ideal stage.

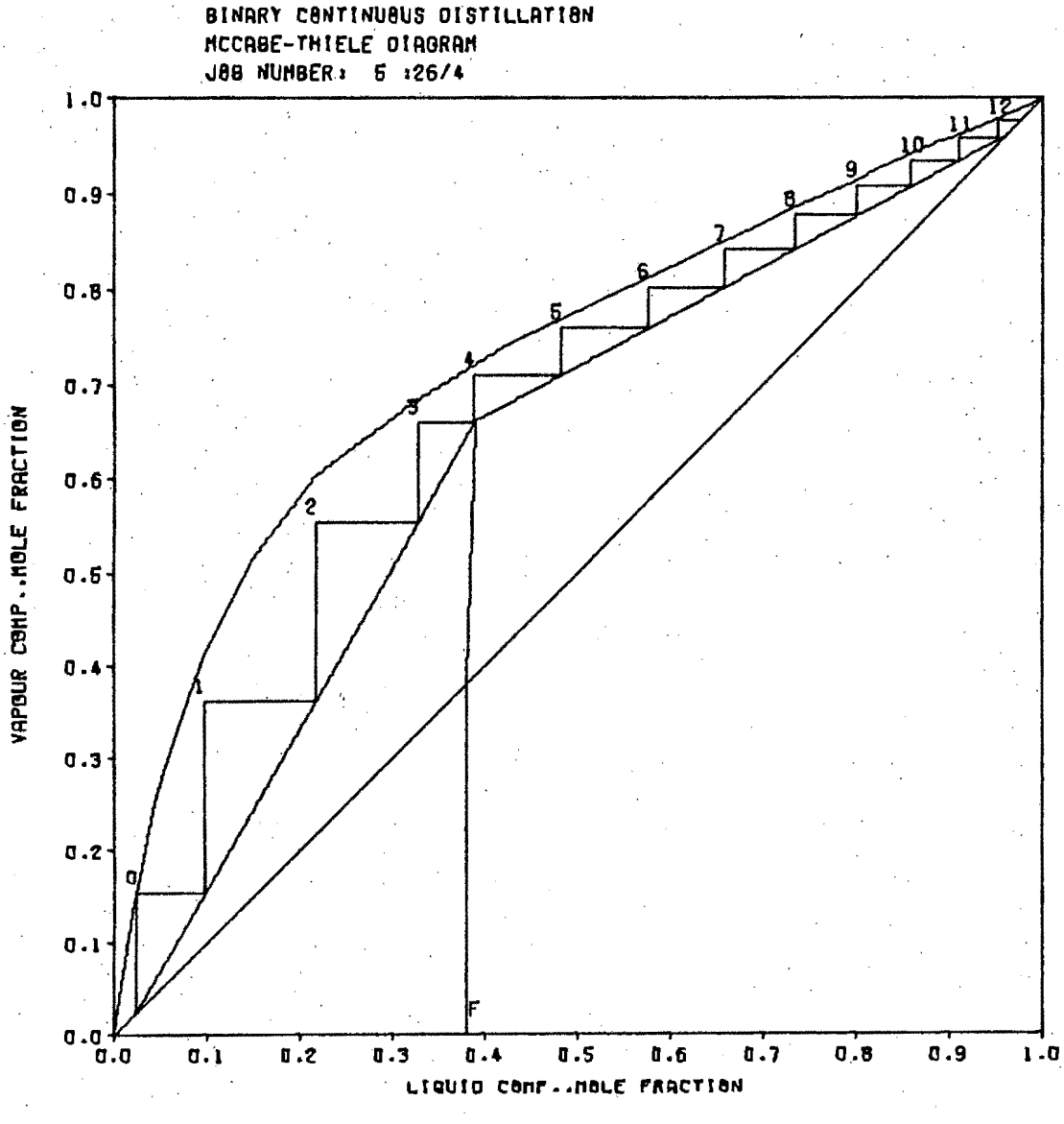


Figure 8.525 Problem 8.521 - Tray efficiency = 0,8.

All diagrams shown in this section are constructed by PLOTTER under code 4 (see section 7.551).

8.522 Ponchon-Savarit problems: Figure 8.526 shows the computer solution to problem 9.8 in Treybal [9] p.326, using Antoine constants. A result table for the same problem using equilibrium data is shown in Table 8.522 below. Agreement with the solution shown in Treybal is good, providing equilibrium data are used. Treybal's results, converted to the SI where necessary, are shown in parenthesis in the table.

TABLE 8.522

SOLUTION TO PROBLEM 9.8 TREYBAL [9] p.326

FEED Q VALUE AND TEMPERATURE DATA:
Q: 1.0183 FEED TEMP: 57.80 DEG C
DEW PT: 89.46 DEG C BUBBLE PT: 76.36 DEG C
(76.1)

SPECIFIED REFLUX RATIO IS LESS THAN THE CALCULATED MINIMUM.
RMIN= .66 R SPEC= .30
A VALUE OF 1.5*RMIN WILL BE USED.

PONCHON-SAVARIT DESIGN OF A BINARY DISTILLATION SYSTEM.

FEED COMPONENTS: (1) METHANOL
(2) WATER

NO. OF THEORETICAL STAGES REQUIRED: 7.4 (7.9)

MURPHREE EFFICIENCY FACTOR USED: 1.00

FEED AND SIDE STREAM SPECIFICATIONS:

STREAM NO	TRAY NO TYPE	Q VALUE	FLOWRATE KGMOLE/HR	COMPOSITION MOL FRAC
1 FEED	4 (4)	1.02 (1.03)	98.10	.3600

FEED TEMPERATURE: 57.80 DEG C (57.77)

TOTAL CONDENSER OPERATION.
CONDENSER PRESSURE: 101.32 KPA
CONDENSER DUTY: 758.67 KW (753.0)

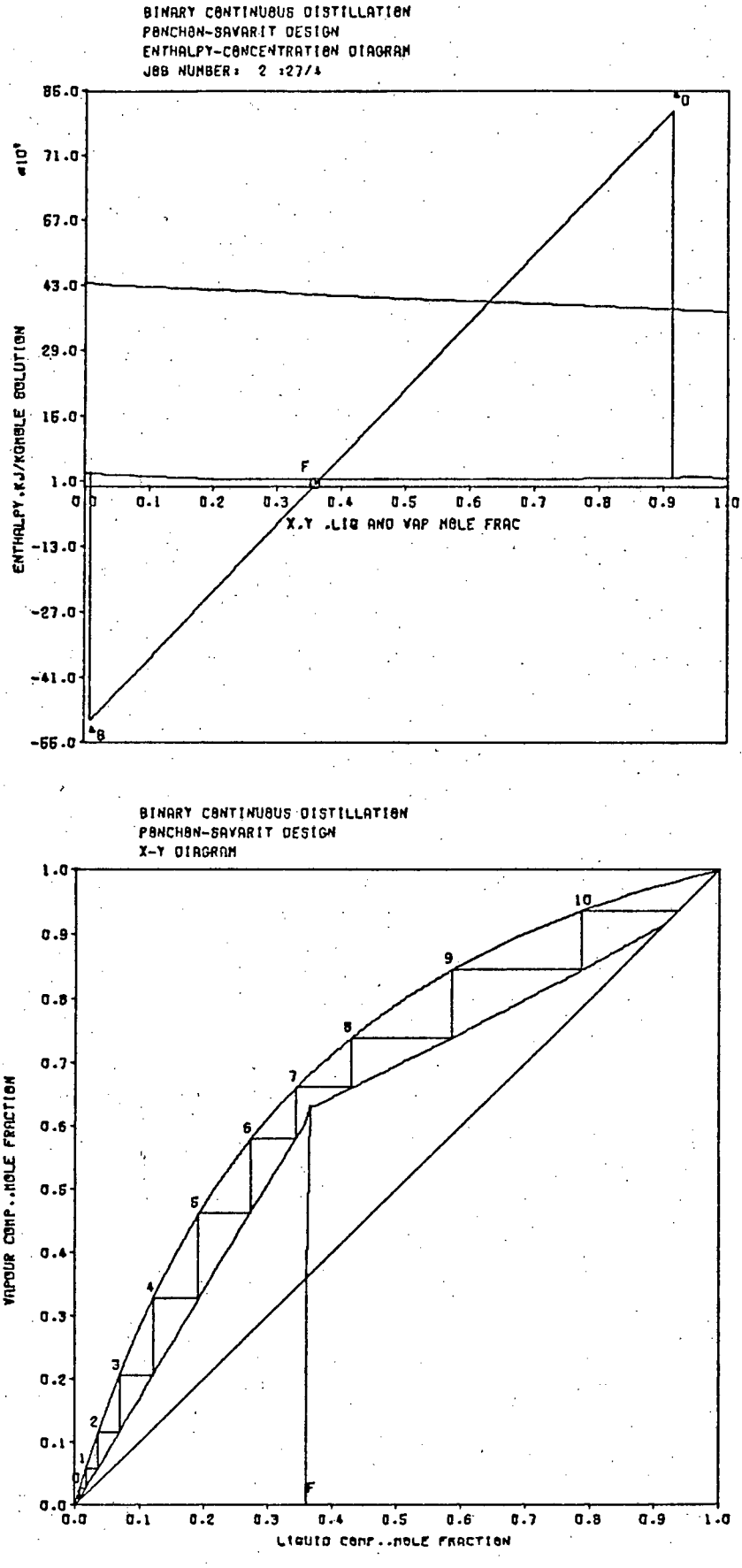


Figure 8.526 Ponchon-Savarit diagram using Antoine constants.

```
REFLUX RATIO:          .9932      (1.029)
MIN REFLUX RATIO:      .6621      (0.685)

DISTILLATE WITHDRAWAL RATE:   38.10 KGMOLE/HR
DISTILLATE COMPOSITION:         .9150

BOTTOMS WITHDRAWAL RATE:    60.00 KGMOLE/HR
BOTTOMS COMPOSITION:          .0076      (0.0056)
REBOILER DUTY:              807.55 KW

COMPOSITION,TEMPERATURE AND FLOWRATE PROFILES:
TRAY    COMPOSITION      TEMPERATURE       FLOWRATES
        (MOLE FRAC)        (DEG C)        (KGMOLE/HR)
        LIQ      VAP     BUBBLE PT        LIQ      VAP
0       .0076    .0076      98.29         60.00    71.57
1       .0339    .0558      94.39        132.14    72.14
2       .1173    .2063      86.56        133.94    73.94
3       .2600    .4566      79.33        137.04    77.04
4       .3598    .6357      76.37        136.75    76.75
FEED TRAY POSITION:
5       .4773    .7058      73.62         34.86    72.96
6       .6139    .7694      70.90         35.68    73.78
7       .7444    .8315      68.44         36.52    74.62
8       .8661    .8908      66.50         37.44    75.54
COND    .9452    .9452      65.70         37.84      .00

ENTHALPY DIAGRAM DEL PTS:
  DEL D=   .735637+05 KJ/KGMOLE      (74595.0)
  DEL B= -.458130+05 KJ/KGMOLE       (-43472.0)
```

The use of Antoine constants lifts the feed position by 3 trays and increases the total number required to 9,8. The minimum reflux ratio is increased to 0,8 and the feed q-value to 1,02. Table 8.522 shows a minimum reflux ratio of 0,66. The reflux ratio was specified as 0,3 to force the program to use a value of 1,5 x R_M, which is the procedure adopted in Treybal. The enthalpy and equilibrium data used by Treybal yield a minimum reflux ratio of 0,685. Using Treybal's resultant reflux ratio of 1,029, the reboiler and condenser heat duties increase as expected, although the total number of trays does not alter much (7,2). The discrepancies in the results emphasize the importance of using realistic enthalpy and equilibrium data in distillation designs.

The condenser compositions shown in the composition-profile table (Table 8.522) refer to the top stage stepped off on the x,y diagram (stage 8). The distillate composition refers to the product from stage 7.4. The enthalpy diagram DEL points are the enthalpy coordinates of Δ_D and Δ_B respectively. These are required by PLOTTER under code 5 (see section 7.552). Tables of operating diagram data may be requested as for McCabe-Thiele problems. The Ponchon-Savarit operating diagram may be plotted manually using these data.

Figure 8.257 demonstrates the use of equilibrium data for a system that deviates appreciably from Raoult's Law (acetone-water). If the feed condition is such that the reboiler stage spans the point of intersection of the operating lines, as shown, the user is informed that the column consists of a rectifying section only. Calculations then continue as normal.

The effects of feed temperature, reflux ratio, condenser type and tray efficiency on Ponchon-Savarit systems can be investigated using BINARY.

8.53 Binary distillation in packed columns - program PACKED

This program uses the procedures described by Treybal [9], who shows only one sample problem. Solving problems manually using this procedure is extremely laborious, which is probably why no further reference to the method could be found. The program PACKED, which is able to solve problems in a matter of seconds, has proved useful in revealing certain discrepancies between results obtained using Shulman's correlations and other design methods. In addition, errors are shown to occur in the calculations of Treybal.

The design of a packed column is apparently more complete than the tray-column design discussed previously, in that the diameter and height of the column are established. Packed column design is based on superficial flowrates and pressure drop, however, therefore diameter and height are necessary features of the calculation procedure.

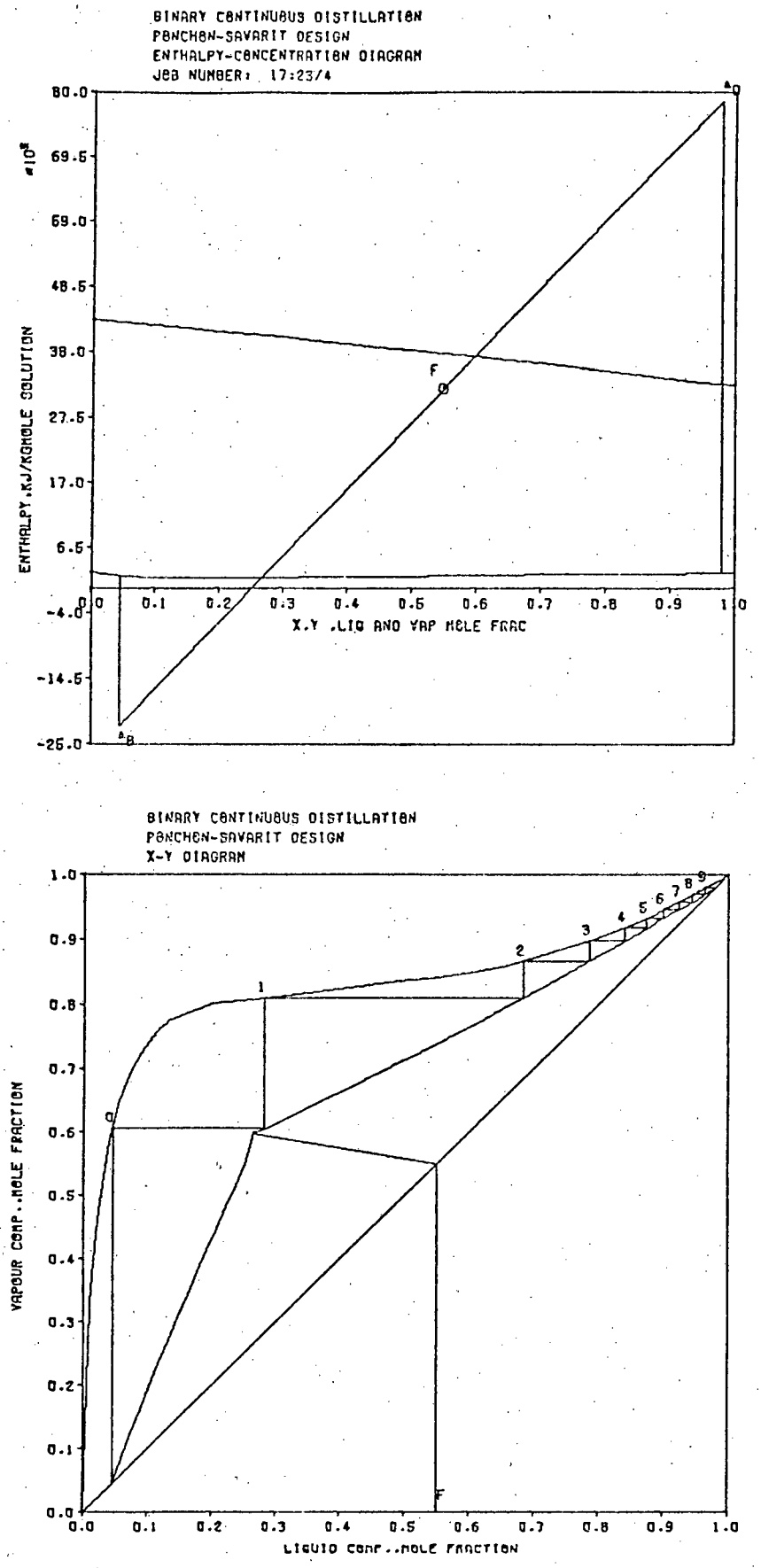


Figure 8.527 Ponchon-Savarit diagram for Acetone - Water system.

In Treybal [9] (p.365), problem 9.12 shows the design of a packed column for the system described previously (Table 8.522). The computer solution, using PACKED, (shown in Table 8.531 below) agrees fairly well with data listed in Treybal for 38,1 mm Raschig rings. Treybal's results are shown in parenthesis.

TABLE 8.531

SOLUTION TO PROBLEM 9.12 TREYBAL [9] p.365

DESIGN OF A PACKED DISTILLATION COLUMN.
MASS TRANSFER COEFFICIENT CALCULATIONS BY SHULMANS METHOD

FEED COMPONENTS: (1) METHANOL
(2) WATER

FEED STREAM SPECIFICATIONS:

FLOWRATE:	98.10	KGMOLE/HR	
COMPOSITION:	.3600	MOLE FRAC	
QVALUE:	1.02		(1.03)
TEMPERATURE:	57.80	DEG C	(57.77)

TOTAL CONDENSER OPERATION.

CONDENSER DUTY:	758.67	KW	(753)
CONDENSER PRESSURE:	101.32	KPA	

REFLUX RATIO:	.9932	(1.029)
MIN REFLUX RATIO:	.6621	(0.685)

DISTILLATE WITHDRAWAL RATE:	38.10	KGMOLE/HR
DISTILLATE COMPOSITION:	.9150	

BOTTOMS WITHDRAWAL RATE:	60.00	KGMOLE/HR	
BOTTOMS COMPOSITION:	.0076		(0.0056)
REBOILER DUTY:	807.55	KW	(824.7)

VAPOUR PRESSURE-DROP AT LOADING: .408 KPA/M (0.408)

THE COLUMN DIAMETER IS BASED ON CONDITIONS AT THE POINT IN THE COLUMN WITH THE GREATEST LIQUID FLOWRATE.
(SEE TABLE BELOW)

RECOMMENDED COLUMN DIAMETER: .790 M (0.75)

HEIGHT OF PACKING REQUIRED:

RECTIFYING SECTION:	5.77	M	(7.4)
STRIPPING SECTION:	2.88	M	(4.48)

```
DO YOU WANT COMPOSITION,TEMPERATURE AND FLOWRATE DATA
TABLES?
NO

DO YOU WANT MASS-TRANSFER DATA TABLES?
YES

PRINCIPAL MASS TRANSFER RESISTANCE IN VAPOUR PHASE.

SHULAMN MASS TRANSFER CALCULATIONS:

  X  =LIQUID MOLE FRACTION OF LIGHT COMPONENT
  V  =VAPOUR RATE IN COLUMN (KGMOLE/HR.SQ M)
  A  =INTERFACIAL SURFACE AREA OF PACKED SECTION (SQ M/CU M)
   KY=VAPOUR MASS TRANSFER COEFFICIENT (KGMOLE/HR.SQ M.MOLE
      FRAC)
  KX =LIQUID MASS TRANSFER COEFFICIENT (KGMOLE/HR.SQ M.MOLE
      FRAC)
```

X	V	A	KY	KX	X	A	KY	KX
.9150	154.926	20.949	7.4256	32.0684	(0.915	19.935	5.492	37.8)
.8601	154.007	20.834	7.3591	31.0438				
.8051	153.137	20.741	7.2957	30.0494				
.7502	152.317	20.673	7.2341	29.0737				
.6953	151.567	20.627	7.1750	28.1515				
.6403	150.854	20.608	7.1179	27.2423	(0.60	21.55	5.56	31.00)
.5854	150.169	20.614	7.0627	26.3676				
.5304	149.483	20.656	7.0068	25.4916				
.4755	148.825	20.732	6.9525	24.6492				
.4206	148.196	20.850	6.8997	23.8242				
.3656	147.606	21.015	6.8483	23.0197	(0.37	21.91	5.565	27.8)
FEED POSITION								
.3656	151.538	34.884	7.0500	42.5693	(0.37	37.99	5.907	51.26)
.3097	159.363	34.429	7.1251	41.7750				
.2537	156.889	32.878	6.9199	40.3227				
.1978	154.398	31.384	6.7257	38.9371	(0.20	32.8	6.093	53.46)
.1418	151.938	29.942	6.5649	37.6656				
.0859	149.465	28.546	6.5004	36.4664				
.0339	147.215	27.270	6.9029	35.5841	(0.02	22.2	5.712	53.95)

Treybal's results indicate a smaller diameter than that designed using PACKED (0.75 m vs. 0.79 m). This is one cause of the greater heights of packing required. The differences in calculation and integration procedure could also contribute to the discrepancies. The effect of specifying a diameter of 0.75 in PACKED is shown in Table 8.532.

In Table 8.531 it is seen that the interfacial areas are in good agreement with the value computed by Treybal whereas the mass transfer coefficients are greater than Treybal's for the vapour phase, and smaller for the liquid phase. From Shulman's correlations it is probable that physical property data affect these calculations.

Treybal states that if the column is redesigned using 38,1 mm Berl saddles the packing height is reduced by roughly 40%. Using PACKED, it is found that almost twice the bed height is required for Berl saddles, although the pressure drop is less for a column of the same diameter.

By inspection of the equations and correlations of Shulman [61] it can be shown that for a given liquid rate the value of H* for Raschig rings is about 60% greater than H* for Berl saddles. The specific interfacial surface area for distillation is given by:

$$a_A = a_{AW}.H^* \qquad \text{(Equation 4.12, 4.15)}$$

It is found that H* dominates the equation and thus although a_{AW} is greater for Berl saddles, a_A is smaller. The end result is that a greater bed height of Berl saddles is required for a given separation. Referring to the original articles of Shulman, the correlations quoted in Treybal appear to be correct.

Furnas and Taylor [79] have shown that for distillation the bed height required is largely independent of the packing material used. Eckert [17] does not discuss the relative efficiencies of packings, but notes that the lower pressure drop of Berl saddles is an advantage with respect to operating costs. Fellinger [80] notes that Berl saddles are up to 25% more efficient than Raschig rings when applied to absorption problems. It appears, therefore, that in practice very little

is known about the behaviour of packing materials in distillation applications. Most designs are carried out using simplified HTU methods [17] based on experimental observations or practical experience.

For tutorial purposes the method of Shulman is useful for demonstrating the effects of component physical properties and other variables on packed column operations. It appears that despite the findings of Treybal, the Shulman correlations do not show that Berl saddles are more efficient than Raschig rings. There does not, however, seem to be conclusive evidence in the literature that Berl saddles are more efficient when applied to distillation hence the results obtained using PACKED are possibly not unreasonable. Experimental work would be required to check the validity of Shulman's correlation when applied to distillation processes.

The ultimate decision inevitably rests on the cost factor. With the present emphasis on reducing operating costs, the cost of extra packing would probably be acceptable when compared with the costs of operating at a higher pressure drop in Raschig rings.

PACKED also shows that the design is sensitive to the equilibrium data used, the reflux ratio and the feed condition. Table 8.532 shows the bed heights under different conditions. Data shown in Table 8.531 applies unless otherwise stated.

These results show:

1) Efficiency and pressure drop are reduced as the size of the packing increases.

2) Higher superficial mass flowrates, caused by reducing the column diameter for a given throughput, improve efficiency.

3) Increasing the reflux ratio reduces the height of packing required, and increases the pressure drop if the diameter is not increased.

TABLE 8.532

PACKED COLUMN DESIGN USING PACKED

Solution	Data Used	Packing Type	Packing Size mm	Column Diameter m	Pressure drop kPa	Z_R m	Z_S m
Treybal	basic	R	38,1	0,75	0,408	7,4	4,48
Program	basic	R	38,1	0,79	0,408	5,77	2,88
"	basic	R	38,1	0,75	0,53	6,02	2,
"	basic	B	38,1	0,72	0,408	10,72	3,40
"	basic	B	38,1	0,79	0,25	10,90	4,49
"	Ant.const.	R	38,1	0,82	0,408	4,37	6,82
"	basic	R	25,4	0,90	0,408	3,38	2,04
"	basic	R	25,4	0,79	0,810	3,81	1,98
"	basic	R	50,8	0,72	0,408	5,62	3,0
"	basic	R	50,8	0,79	0,25	5,31	3,04
"	R=2,5	R	38,1	1,01	0,408	4,28	2,12
"	R=2,5	R	38,1	0,79	1,6	3,94	1,80
"	q=0,68	R	38,1	0,64	0,408	6,27	2,58
"	q=0,68	R	38,1	0,79	0,15	6,18	2,95

Table codes:

Packing Type: R - Raschig rings; B = Berl saddles

Data used: basic - data specified for problem 9.11 in Treybal [9] p.365.
Ant.const.- basic with Antoine constants used to generate equilibrium data.
R - basic with reflux ratio changes.
q - basic with feed temperature changed.

Pressure drop: from Figure 1, Appendix A.

Z_R, Z_S : height of packing in rectifying and stripping section respectively.

---

4) Increasing the feed temperature reduces the pressure drop and the efficiency for a fixed diameter and throughput. These points are in agreement with general experience [17,63, 79].

It can also be seen that if Antoine constants are used a greater height of packing is required in the stripping section. This is equivalent to raising the feed position in a tray column as shown in section 8.522.

8.54 Multicomponent distillation - program MULCOM

The time required to solve multicomponent distillation problems manually has lead to an emphasis on short-cut techniques in the undergraduate course. MULCOM, like PACKED, is able to solve complex problems in a matter of seconds, thus enabling students to investigate the effects of several parameters on multicomponent systems.

Solutions to three problems shown below demonstrate the capabilities of MULCOM.

Problem 8.541.

(Data from Example 9.13, Treybal [9] p.388)

The system considered consists of a six-component hydrocarbon system in a column containing 14 trays. Table 8.541 shows the abbreviated result table printed on the terminal in demand mode executions. Table 8.542 shows two sections of the detailed result table that may be obtained via PRINTFILE (see section 7.422). The full table shows similar data for each tray in the column, and overall material and enthalpy balance figures. Treybal's results are shown in parenthesis in Table 8.541. The distillate composition and temperature profiles are in close agreement. The discrepancies in the heat duties are due to the enthalpy data chosen.

TABLE 8.541

SOLUTION TO EXAMPLE 9.13, TREYBAL [9] p.372

MULTICOMPONENT DISTILLATION BY THIELE-GEDDES METHOD:

RUN NO: 3/15/ 3

OVERALL MATERIAL BALANCE:

STREAM	FLOWRATE (KGMOLE/HR)
DISTILLATE	38.4600
BOTTOMS	61.5400
FEED-TOTAL	100.0000
-VAPOUR	37.1501
-LIQUID	62.8499

COMPONENT MOLE FRACTIONS:

COMP.	DISTILLATE	BOTTOMS	FEED STREAM LIQUID	FEED STREAM VAPOUR	FEED STREAM TOTAL
1	.0780 (0.0789)	.0000	.0019	.0775	.0300
2	.1820 (0.1842)	.0000	.0170	.1596	.0700
3	.3885 (0.3870)	.0010	.0860	.2582	.1500
4	.3481 (0.3420)	.3187	.3286	.3323	.3300
5	.0035 (0.0079)	.4853	.3907	.1466	.3000
6	.0000 (0.0000)	.1750	.1757	.0258	.1200

TEMPERATURE AND PRESSURE PROFILE:

TRAY NO	TEMPERATURE DEG C	TEMPERATURE DEG F	PRESSURE KPA
CONDENSER	51.0	123.8 (115)	1034.21
14	62.7	144.9 (138)	1034.21
13	66.4	151.6 (150)	1034.21
12	69.4	157.0 (158)	1034.21
11	73.1	163.7 (165)	1034.21
10	78.1	172.5 (176)	1034.21
FEED	84.6	184.2 (188)	1034.21
8	92.1	197.8 (202)	1034.21
7	96.6	206.0 (206)	1034.21
6	98.6	209.6 (208)	1034.21
5	100.1	212.2 (210)	1034.21
4	101.2	214.2 (212)	1034.21
3	102.2	215.9 (217)	1034.21
2	103.3	218.0 (221)	1034.21
1	105.7	222.3 (227)	1034.21
REBOILER	112.2	234.0 (234)	1034.21

NO. OF ITERATIONS= 14 THETA= 1.000

REFLUX RATIO= .80
DISTILLATE RATE= 38.46 KGMOLE/HR
FEED Q VALUE= .6285 (0.67)
FEED TEMPERATURE= 82.20 DEG C (82.2)
FEED ENTHALPY= 11798.34 KJ/KGMOLE

CONDENSER DUTY= 153.57 KW (74.5)
REBOILER DUTY= 206.38 KW (108,5)

PARTIAL CONDENSER OPERATION.

VAPOUR AND LIQUID FLOWRATES LEAVING TRAYS:

TRAY NO	VAPOUR	LIQUID	(KGMOLE/HR)
CONDENSER	38.46	30.77	
14	69.23	31.16	
13	69.62	30.81	
12	69.27	30.26	

11	68.72	29.60
10	68.06	28.98
FEED	67.44	93.93
8	32.39	99.18
7	37.64	95.99
6	34.45	97.59
5	36.05	98.82
4	37.28	99.71
3	38.17	100.30
2	38.76	100.58
1	39.04	100.39
REBOILER	38.85	61.54

Figure 8.541 shows the corresponding diagrams produced by PLOTTER (code 1). The temperature profile diagram shows profiles for three runs at different reflux ratios.

RUN	REFLUX RATIO
1	0,8
2	8,0
3	2,5

The remaining diagrams correspond to run 1 which is equivalent to the example in Treybal.

Problem 8.542

A benzene-toluene-xylene column is considered. Figure 8.542 shows the composition profiles for the same system under the following conditions:

a) Assuming ideal trays;
b) Using a single set of component efficiency data (section 7.434);
c) Applying Nord's [81] efficiency data for the B-T-X system.

The efficiency data used in (b) is not representative of the true conditions, but is useful for demonstrating the general effect of efficiency specifications. Efficiency data compiled experimentally, or from a source such as Nord, are probably more representative, but such data are seldom available for

TABLE 8.542

CONDENSER AND FEED TRAY SECTIONS OF THE DETAILED RESULT TABLE.

PARTIAL CONDENSER

COMPONENT NO	VAPOUR DISTILLATE KGMOLE/HR	VAPOUR DISTILLATE MOL FRACTION	LIQUID REFLUX KGMOLE/HR	LIQUID REFLUX MOL FRACTION
1	3.00	.780631-01	2.40	.251282-02
2	7.30	.182007+00	5.60	.305821-01
3	14.94	.386522+00	11.95	.234108+00
4	13.38	.348024+00	10.71	.711053+00
5	.13	.344259-02	.11	.217148-01
6	.00	.156456-05	.00	.289260-04
TOTALS:	38.46	1.0000	30.77	1.0000

CONDENSER TEMPERATURE= 50.97 DEG C

CONDENSER PRESSURE= 1034.21 KPA

TRAY NO 9 • FEED TRAY •

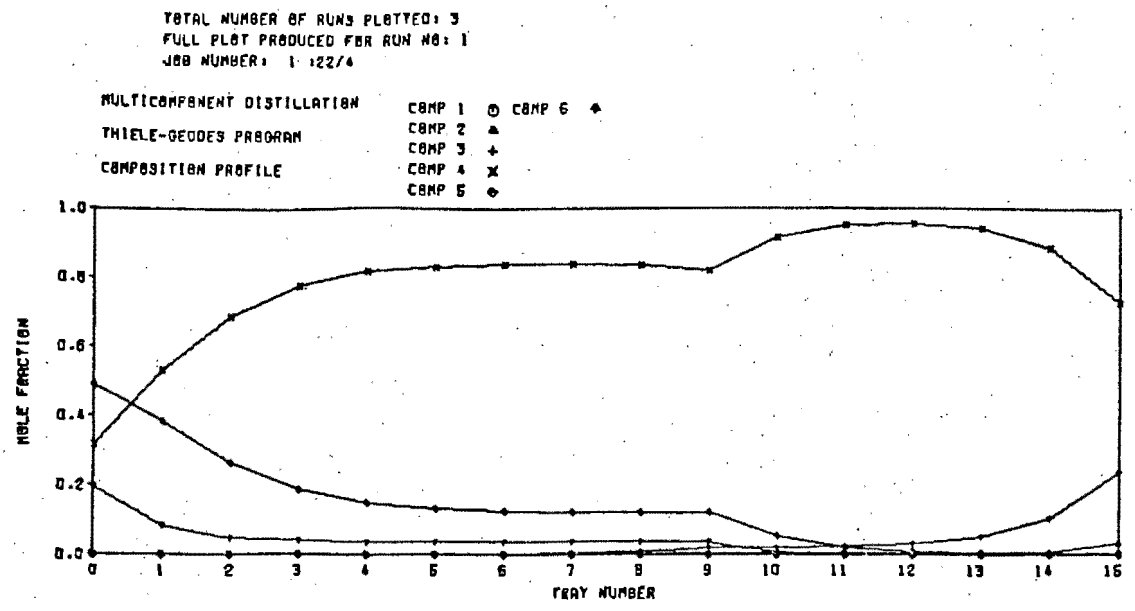
COMPONENT NO	VAPOUR FEED KGMOLE/HR	VAPOUR FEED MOL FRACTION	LIQUID FEED KGMOLE/HR	LIQUID FEED MOL FRACTION	VAPOUR LEAVING KGMOLE/HR	VAPOUR LEAVING MOL FRACTION	LIQUID LEAVING KGMOLE/HR	LIQUID LEAVING MOL FRACTION
1	2.86	.777563-01	.12	.195015-02	3.03	.449743-01	.10	.110480-02
2	5.92	.160377+00	1.08	.171384-01	7.35	.109046+00	1.06	.112809-01
3	9.55	.258600+00	5.45	.864352-01	17.70	.262380+00	7.92	.840492-01
4	12.24	.331617+00	20.78	.329065+00	27.76	.411573+00	36.55	.387687+00
5	5.38	.145774+00	24.62	.390220+00	10.16	.150677+00	35.75	.379474+00
6	.94	.255953-01	11.06	.175223+00	1.44	.213490-01	12.83	.136204+00
TOTALS:	36.91	1.0000	63.09	1.0000	67.45	1.0000	94.22	1.0000

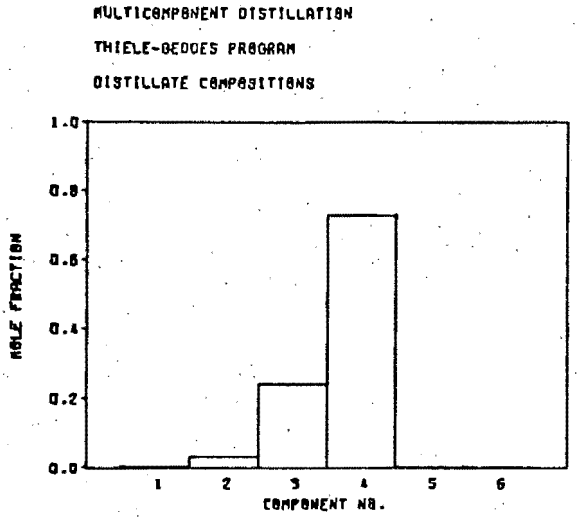
TRAY TEMPERATURE= 84.51 DEG C

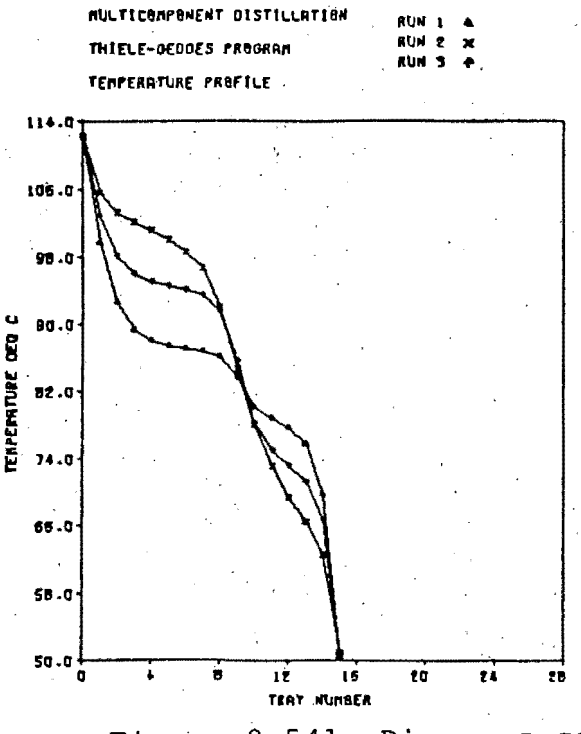
TRAY PRESSURE= 1034.21 KPA

FEED Q VALUE= .63

FEED TEMPERATURE= 82.04 DEG C







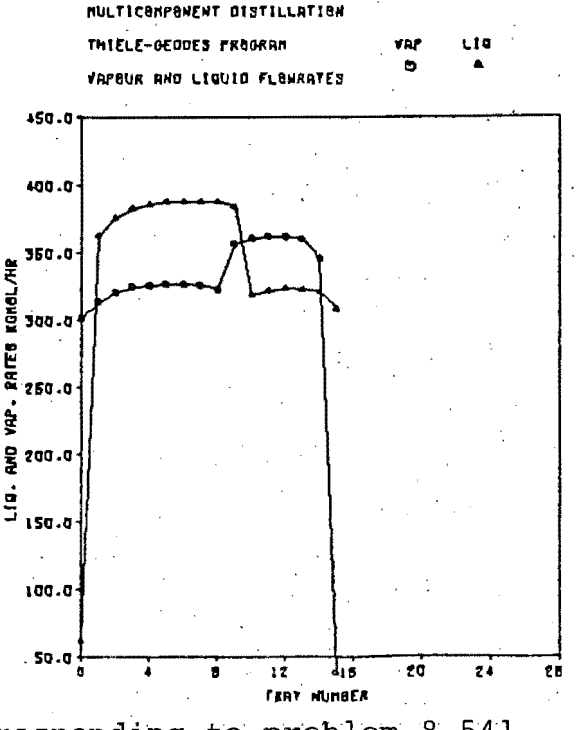
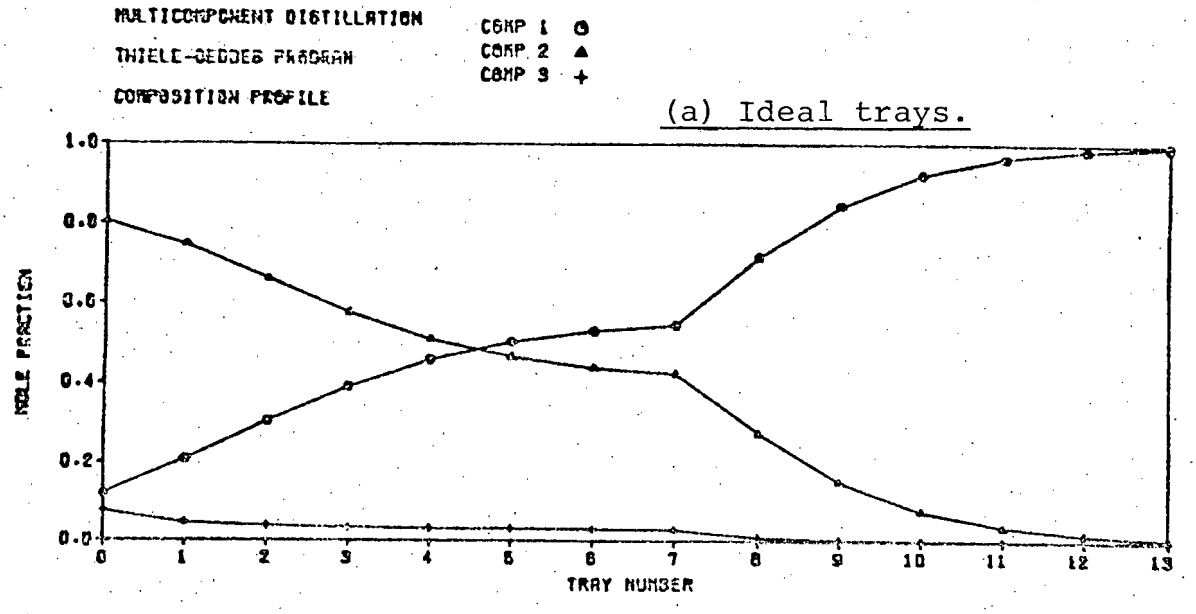
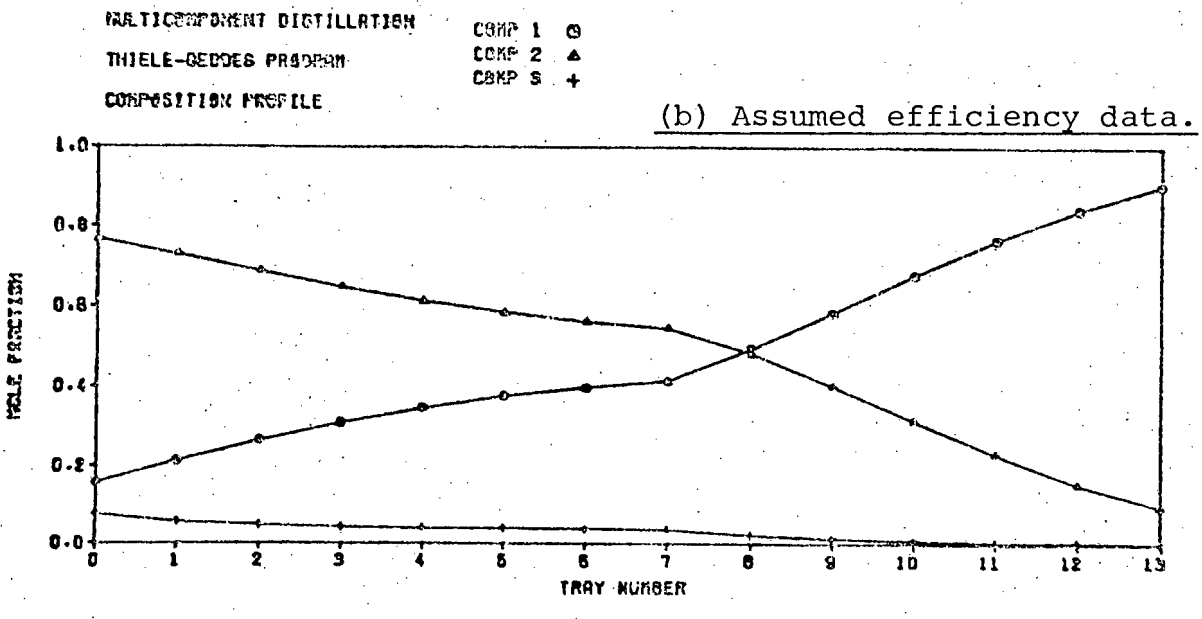


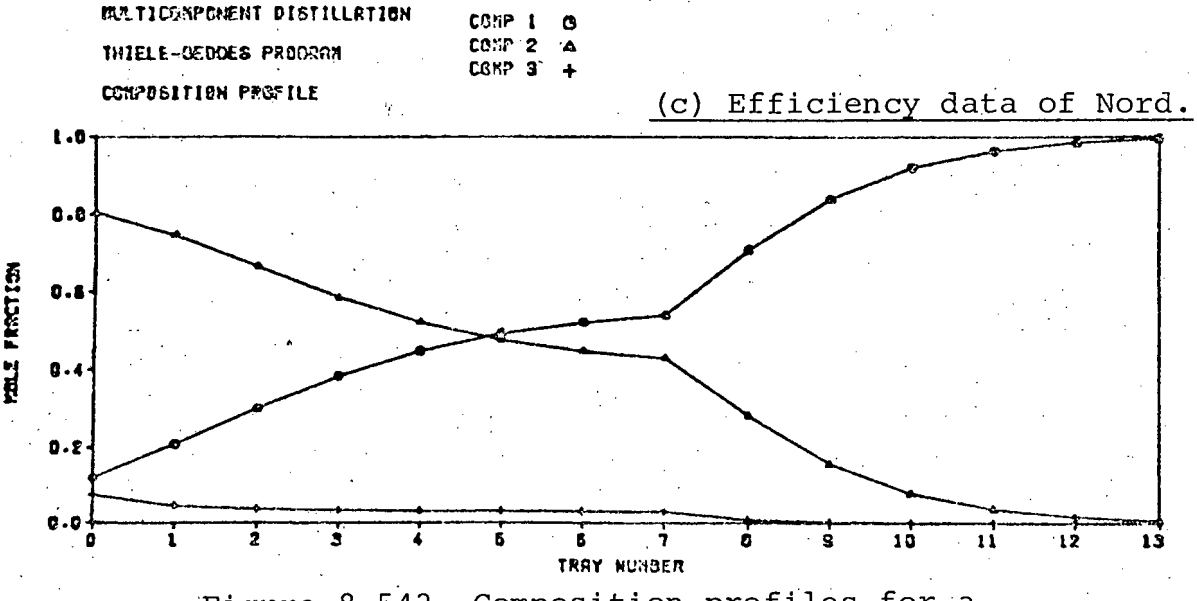
Figure 8.541 Diagrams corresponding to problem 8.541.



(a) Ideal trays.



(b) Assumed efficiency data.



(c) Efficiency data of Nord.

Figure 8.542 Composition profiles for a Benzene-Toluene-Xylene system.

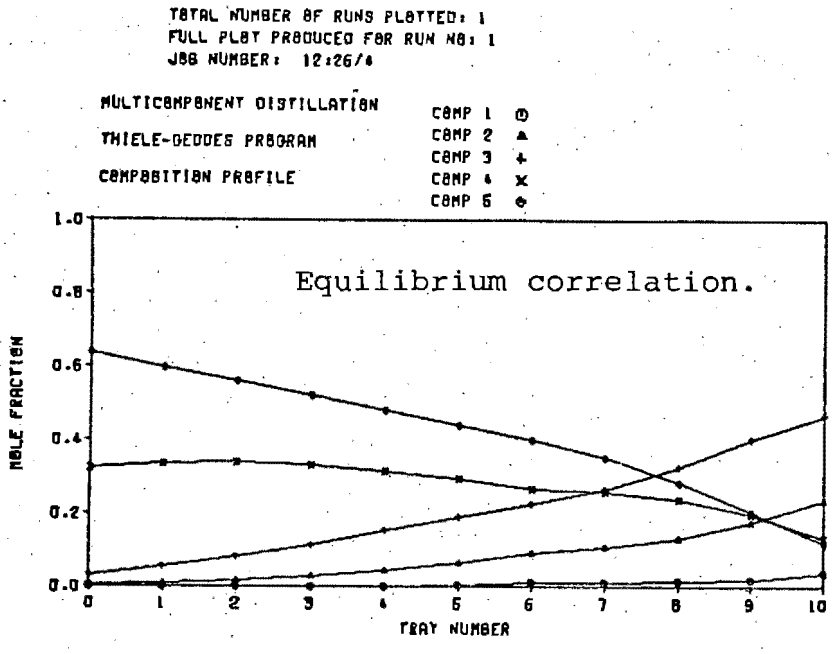
tutorial problems.

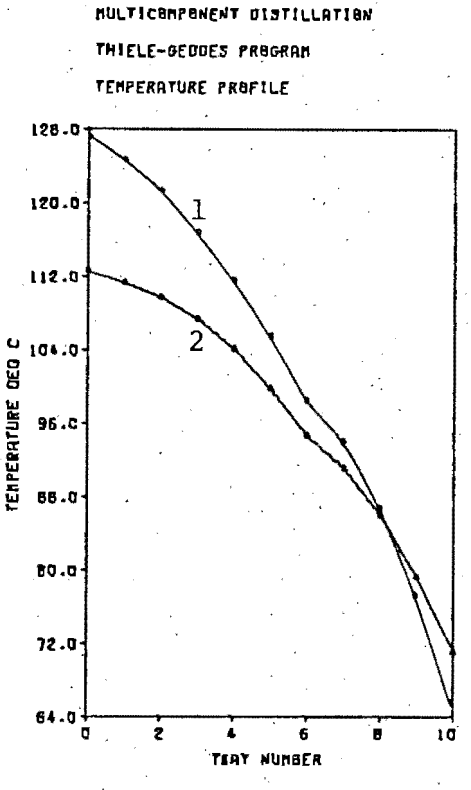
The data of Nord cause very little change in the concentration profiles. If efficiency factors are assumed, as in (b), the effect is more noticeable but less realistic.

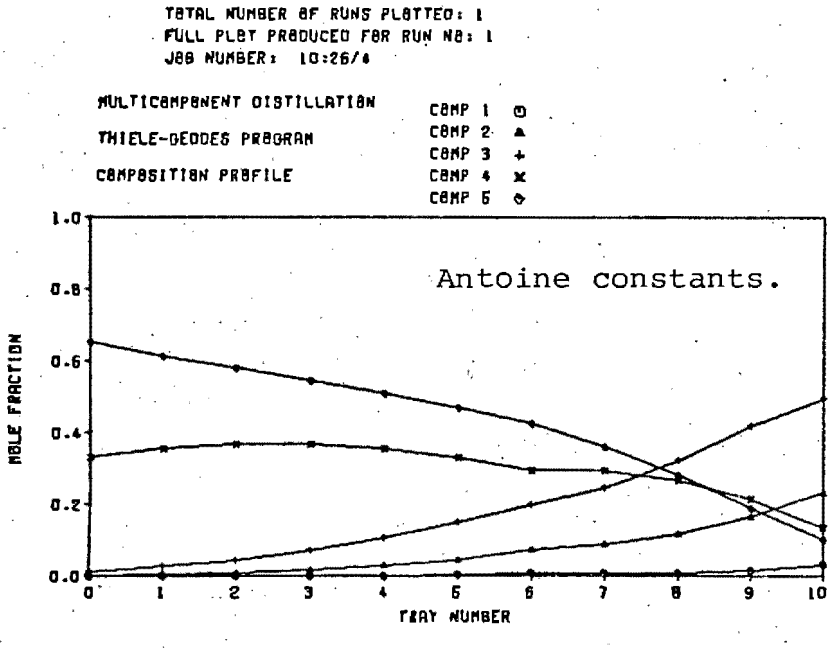
Problem 8.543

This problem, taken from Holland [21], involves a five component hydrocarbon system. It is used here to compare the use of Antoine constants and equilibrium constant polynomials (section 7.433). Although polynomials in temperature are widely used in industry to correlate equilibrium data, the procedure is cumbersome for setting tutorial problems. Antoine constants are readily available for a wide variety of compounds and are thus easier to use. Figure 8.543 shows the composition and temperature profiles for the two systems. The temperature profiles are superimposed for ease of comparison. It is noticeable that although the composition profiles are very similar, the temperature profile generated by Antoine constants is shallower than that for the temperature polynomials. For standard tutorial problems this is not a serious disadvantage, because the overall behaviour of the system is representative of a real column. The enthalpy data correlation corresponding to NENTH=2 (see section 7.432) was used for both runs. Coefficients were taken from Holland [21].

The effects of feed temperature and pressure, equimolar overflow assumptions, and partial or total condenser operation can also be demonstrated using MULCOM.







1 - Equilibrium correlation.

2 - Antoine constants.

Figure 8.543 Diagrams corresponding to problem 8.543.

8.6 CONCLUSIONS AND RECOMMENDATIONS

The objectives of this study have been realized in that an integrated computer package has been developed, capable of solving tutorial problems covering all aspects of distillation discussed in the undergraduate course. The programs are generally more economical both in computer time and in core requirements than comparable programs published by CACHE [13]. The programs may be operated both in demand and in batch mode, although the conversational user-program interaction can only be utilised effectively in demand mode.

The design of packed columns is more complete in that the dimensions of the column are determined. This, however, is an integral part of a packed column design, which is based on superficial flowrates and pressure drop considerations. The detailed design of column components was not included in this study and is not necessary for studying the performance of tray columns at an elementary level.

The programs include a number of options that reduce the user's task in compiling data tables and solving problems. The plotting facility is useful for investigating the effects of variables on Ponchon-Savarit diagrams, and multicomponent concentration profiles. These diagrams are normally time consuming to construct manually. Programs for solving Ponchon-Savarit systems, and for designing packed distillation columns do not appear to have been published before. The CACHE programs do not consider enthalpy effects in binary systems. The program PACKED has proved useful in revealing certain errors in published sample solutions. This has suggested that the effect of holdup on the performance of packing materials should be investigated more thoroughly.

The programs have been designed to facilitate alterations and additions in the future. The addition of a tray design program using data directly from MULCOM would complete the

multicomponent design facility. Programs dealing with liquid-liquid extraction and gas absorption should be added to the package. The package would then cover all mass transfer operations considered in the undergraduate course. The following subroutines are designed for general use in future programs:

(1) Bubble point and dew point calculations (BINBPT, BINDPT,BUBCAL,DEWCAL).
(2) Flash calculation routines (FLASHB and QFLASH).
(3) Iterative search procedure (SEARCH).
(4) Simpson's rule integration (SIMPS).
(5) Least squares fit of polynomials (FDATA).
(6) Physical property data package. (Section 4.55)

The present package could be used to conduct experimental investigations on packed and tray columns for distillation purposes, particularly with respect to stage efficiencies. As this is primarily a teaching package, however, emphasis has been placed on the following points:

(1) Demonstration of the basic principles of distillation.
(2) Conformity with standard conventions and nomenclature.
(3) Minimising the effort required to set up and solve tutorial problems.
(4) Conversational user interaction and descriptive error messages.
(5) Minimising computer time and core requirements.

The modular structure of the programs should facilitate future additions or alterations to the basic package.

REFERENCES

1. Lewis, W.K. and Matheson, G.L., "Design of rectifying columns for natural and refinery gasoline", Ind.Eng. Chem., 24, 5, 494-498, (1932).

2. Thiele, E.W. and Geddes, R.L., "Computation of distillation apparatus for hydrocarbon mixtures", Ind.Eng. Chem., 25, 3, 289-295, (1933).

3. Robinson, C.S. and Gilliland, E.R., "Elements of Fractional Distillation", 4th ed., McGraw-Hill, New York, (1950).

4. Amundson, N.R. and Pontinen, A.J., "Multicomponent distillation calculations on a large digital computer", Ind.Eng.Chem., 50, 5, 730-736, (1958).

5. Ellis, S.R.M. and Boyes, A.P., "Research and development in distillation: Some aspects of the future", Trans.Inst. Chem.Eng., 52, 202-210, (1974).

6. Zuiderweg, F.J., "Distillation-science and business", Chem.Eng.(Lond), 277, Sept.,404-410, (1973).

7. McCabe, W.L. and Thiele, E.W., Ind.Eng.Chem., 17, 605, (1925) (from Smith [11]).

8. Ponchon, M., Tech.moderne, 13, 20, 55, (1921) (from Smith [11]); Savarit, R., Arts et métiers, (1922) (from Smith [11]).

9. Treybal, R.E., "Mass Transfer Operations", 2nd ed., McGraw-Hill, New York, (1968).

10. McCabe, W.L. and Smith, J.C., "Unit Operations of Chemical Engineering", 2nd ed., McGraw-Hill, New York, (1967).

11. Smith, B.D., "Design of Equilibrium Stage Processes", McGraw-Hill, New York, (1963).

12. Brian, P.L.T., "Stages Cascades in Chemical Processing", Prentice Hall, Englewood Cliffs, N.J., (1972).

13. The CACHE Committee, "Computer Programs for Chemical Engineering-Stagewise Computation", Christensen, J. (Editor).

14. Friday, J.R. and Smith, B.D., "An analysis of the equilibrium stage separations problem-formulation and convergence", A.I.Ch.E.J., 10, 5, 698-707, (1964).

15. Koppel, P.M., "Fast way to solve problems for batch distillations", Chem.Eng.(NY), Oct.16, 109-112, (1972).

16. Strangio, V.A. and Treybal, R.E., "Reflux-stages relations for distillation", Ind.Eng.Chem.Process Des.Dev., 13, 3, 279-285, (1974).

17. Eckert, J.S., "How tower packings behave", Chem.Eng.(NY), Apr.14, 70-76, (1975).

18. Seppala, R.E. and Luus, R., "Evaluation and improvement of simulation procedures for multicomponent distillation", J. Franklin Inst., 293, 5, 325-344, (1972).

19. Van Winkle, M. and Todd, W.G., "Optimum fractionation design by simple graphical methods", Chem.Eng.(NY), 78, Sept.20, 136-148, (1971).

20. Guerreri, G., Peri, B. and Seneci, F., "Comparing distillation designs", Hydrocarbon Process, 51, 12, 77-78, (1972).

21. Holland, C.D. "Multicomponent Distillation", Prentice Hall, Englewood Cliffs, N.J., (1963).

22. Biddulph, M.W., Private Communication.

23. Miskin, L.G., Ozalp, U. and Ellis, S.R.M., "Ternary component efficiencies", Brit.Chem.Eng. and Proc.Tech., 17, 2, 153-155, (1972).

24. Sargent, R.W.H. and Murtagh, B.A., "The design of plate distillation columns for multicomponent mixtures", Trans. Inst.Chem.Eng., 47, T85-T94, (1969).

25. Wang, J.C. and Henke, G.E., "Tridiagonal matrix for distillation", Hydrocarbon Process, 45, 8, 155-163, (1966).

26. Greenstadt, J., Bard, Y. and Morse, B., "Multicomponent distillation calculation on the IBM 704", Ind.Eng.Chem., 50, 11, 1644-1647, (1958).

27. Naphtali, L.M. and Sandholm, D.P., "Multicomponent separation calculations by linearization", A.I.Ch.E.J., 17, 1, 148-153, (1971).

28. Ishii, Y. and Otto, F.D., "A general algorithm for multistage multicomponent separation calculations", Can.J.Chem. Eng., 51, Oct., 601-606, (1973).

29. Rose, A., Sweeny, R.F. and Schrodt, V.N., "Continuous distillation calculations by relaxation method", Ind.Eng. Chem., 50, 5, 737-740, (1958).

30. Davies, R.H., Ph.D. Thesis, University of London, (1963) (from Sargent [24]).

31. Petryschuk, W.F. and Johnson, A.I., "A simulation and parametric study of four existing multicomponent distillation columns", Can.J.Chem.Eng., 44, Oct., 241-251,(1966).

32. Bonner, J.S., "Solution of multicomponent distillation problems on a stored-program computer", Proc.Am.Petroleum Inst., 36, 3, 238-243, (1956).

33. Lyster, W.N., Sullivan, S.L., Billingsley, D.S. and Holland, C.D., "Figure distillation this new way: Part 1", Petr.Refnr., 38, 6, 221-230, (1959).

34. Billingsley, D.S., "On the equations of Holland in the solution of problems in multicomponent distillation", IBM J.Res.Dev., 14, Jan., 33-40, (1970).

35. Holland, C.D. and Pendon, G.P., "Solve more distillation problems: Part 1", Hydrocarbon Process, 53, 7, 148-156, (1974).

36. Newman, J.S., "Temperature computed for distillation", Hydrocarbon Process, 42, 4, 141-144, (1963).

37. Boynton, G.W., "Iteration solves distillation", Hydrocarbon Process, 49, 1, 153-156, (1970).

38. Goldstein, R.P. and Stanfield, R.B., "Flexible method for the solution of distillation design problems using the Newton-Raphson technique", Ind.Eng..Chem.Process Des.Dev., 9, 1, 78-84, (1970).

39. Rowlinson, J.S., "The use of the principle of corresponding states", Chem.Eng.(Lond), 291, Nov., 718-721, (1974).

40. Murphree, E.V., Ind.Eng.Chem., 17, 747, (1925)(from Smith [11].

41. Holland, C.D. and McMahon, K.S., "Comparison of vaporization efficiencies with Murphree-type efficiencies in distillation", Chem.Eng.Sci., 25, 431-436, (1970).

42. McAdams, W.H., Unpublished Notes on Distillation, M.I.T. (from Perry [58], 3rd ed.).

43. Carey, J.S., Sc.D. Thesis, M.I.T. (1930) (from Perry [58], 3rd ed.).

44. Yamada, I., Holland, C.D., Hiraoka, S. and Homma, N., "Distillation point efficiencies by use of thermal effects", J.Chem.Eng.Jap., 6, 6, 512-517, (1974).

45. Yamada, I., Matsumoto, N. and Suzuki, M., "Distillation calculation methods with plate efficiencies", J.Chem.Eng. Jap., 6, 1, 68-73, (1973).

46. Yamada, I., Matsumoto, N. and Holland, C.D., "Approximation for prediction of vaporization plate efficiency", J.Chem.Eng.Jap., 6, 5, 479-462, (1963).

47. Onda, K., Takeuchi, H. and Takahashi, K., "Column efficiencies of rectifying column with liquid mixing on plate", J.Chem.Eng.Jap., 5, 1, 13-19, (1972).

48. MacFarland, S.A., Sigmund, P.M. and Van Winkle, M., "Predict distillation efficiency", Hydrocarbon Process, 51, 7, 111-114, (1972).

49. Asbjørnsen, O.A., "The stage efficiency in dynamic models of phase separation processes", Chem.Eng.Sci., 28, 2223-2229, (1973).

50. Standart, G.L. "Some problems of predicting plate efficiencies", Chem.Eng.(Lond), 291, Nov. 716-618, (1974).

51. Biddulph, M.W., "Multicomponent distillation simulation-distillation of air", A.E.Ch.E.J., 21, 2, 327-335, (1975).

52. Treybal, R.E.,"Mass Transfer Operations", 1st ed., McGraw-Hill, New York, (1955).

53. Bogart, M.J.P., Trans.Am.Inst.Chem.Eng., 33, 139, (1937), (from Perry [58]).

54. Smoker, E.H. and Rose, A., Trans.Am.Inst.Chem.Eng., 36, 285, (1940) (from Perry [58]).

55. Kobe, K.A. "Thermochemistry for the petrochemical industry", Petr.Refnr., Jan. 1949-Nov. 1954.

56. Watson, Ind. and Eng.Chem., 35, 398, (1943) (from Perry [58]).

57. Hougen, O.A., Watson, K.M. and Ragatz, R.A., "Chemical Process Principles, Part II", 2nd ed., Wiley, New York, (1959).

58. Perry, J.H., "Chemical Engineers' Handbook", 5th ed., McGraw-Hill, New York, (1973).

59. Peters, M.S. and Timmerhaus, K.D., "Plant Design and Economics for Chemical Engineers", 2nd ed., McGraw-Hill, New York, (1968).

60. King, C.J., "Separation Processes", McGraw-Hill, New York, (1971).

61. Shulman, H.L., "Performance of packed columns: Part 1, 2 and 3", A.E.Ch.E.J., 1, 2, 247-264, (1955).

62. Van Winkle, M.,"Distillation", McGraw-Hill, New York, (1967).

63. Leva, M., "Tower packings and packed tower design", 2nd ed., U.S. Stoneware Co., Akron, (1953).

64. Sawistowski, H. and Smith, W., "Mass Transfer Process Calculations", Interscience, London, (1963).

65. Brooks, P.C., Hutron, B.E.T., Leung, L.S. and Nicklin, D.J., "Flooding packed towers for more capacity", Chem. Eng.(NY), 81, Feb.18, 152, (1974).

66. Shulman and Margolis, A.E.Ch.E.J., 3, 157, (1957)(from Treybal [9] and Perry [58]).

67. Goldhammer, Z.physik.Chem., 71, 577, (1910)(from Perry [58]).

68. Fishtine, Ind.Eng.Chem.Fund. 2, 149, (1963)(from Perry [58]).

69. Stakhorsky, Z.physik.Chem., 64, 385, (1908), from Gambill, W.R., "Estimate Engineering Properties, Part IV", Reprint from Chem.Eng.(NY)(1958).

70. Lovgren, Svensk Kem.Tid., 53, 359, (1941)(from Gambill, Part IV, [69]).

71. Wilke and Chang, A.E.Ch.E.J., 1, 264, (1955)(from Perry [58]).

72. Arnold, Ind.Eng.Chem., 22, 1091, (1930)(from Gambill) Part IV, [69]).

73. Wilke and Lee, Ind.Eng.Chem., 47, 1253, (1955)(from Perry [58]).

74. Kendall and Monroe, J.Am.Chem.Soc., 39, 1787, (1917) and 43, 115, (1921)(from Perry [58]).

75. Thomas, J.Am.Chem.Soc., 573, (1946)(from Gambill, Part V [69]).

76. Herning and Zipperer, Gas-u. Wasserfach, 79, 49, (1936) (from Gambill, Part V, [69]).

77. Arnold, J.Chem.Phys., 1, 170, (1933) (from Perry [58]).

78. Subcommittee 12, A.I.Ch.E. Report, Chem.Eng.Prog., June, (1956) (from Smith [11]).

79. Furnas, C.C. and Taylor, M.L. "Distillation in packed columns", Trans.Am.Inst.Chem.Eng., 36, 135-169, (1940).

80. Fellinger, Sc.D. Thesis, M.I.T., (1914) (from Perry [58].

81. Nord, M., "Plate efficiencies of benzene-toluene-xylene systems in distillation", Trans.Am.Inst.Chem.Eng., 42, 863-881, (1946).

APPENDIX A

PACKED COLUMN CORRELATION DATA

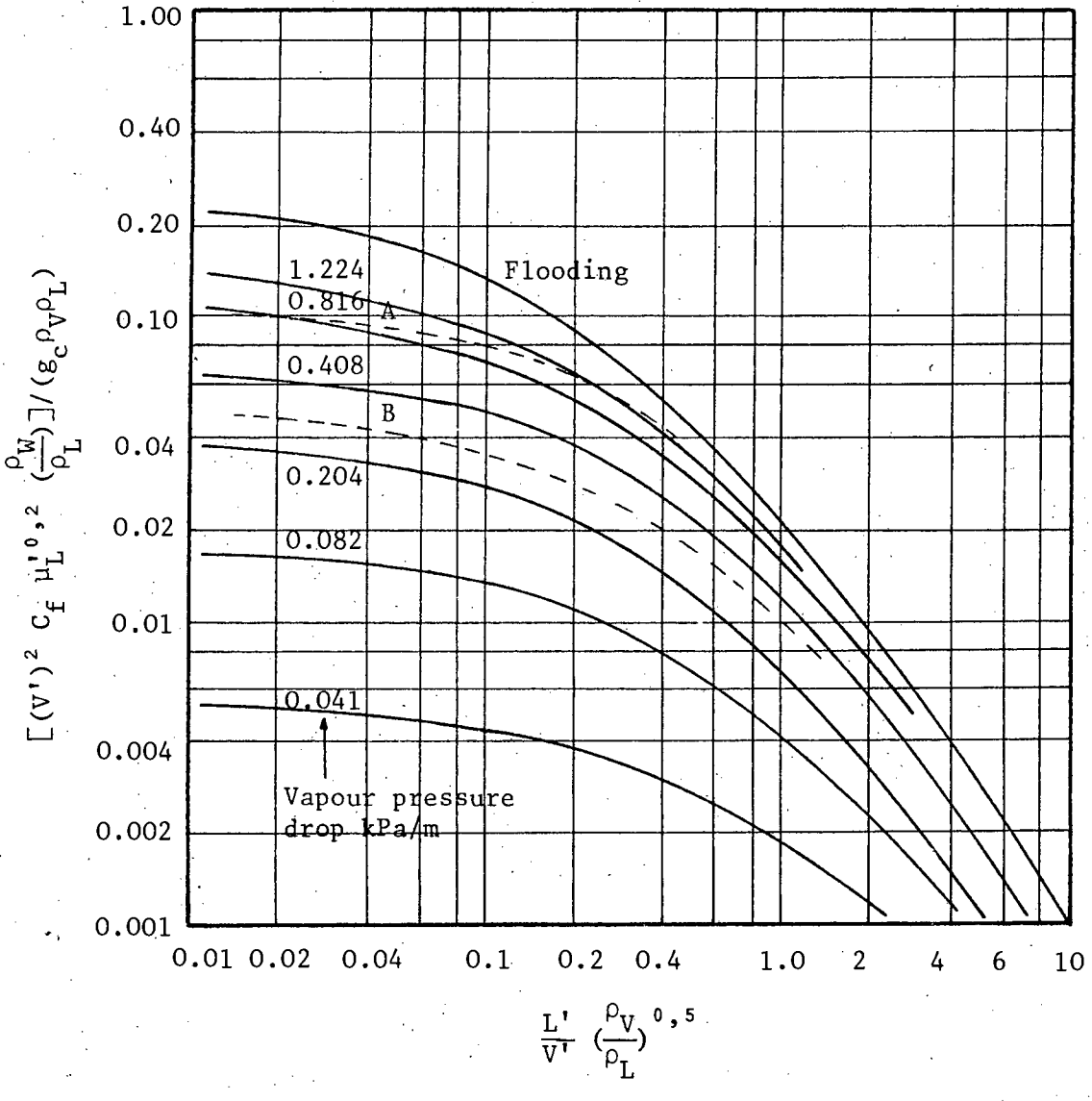


FIGURE 1 FLOODING AND PRESSURE DROP IN PACKED TOWERS

(from Treybal [9], p.160)

Broken lines:

A - upper limit of loading zone [62]

B - lower limit of loading zone [62]

Data used in curve fitting routine FDATA are taken from this diagram and stored in the following data elements. The data elements are accessed by FDATA through the ERTRAN facility.

kPa/m	Element
0.041	FLOOD1
0.084	FLOOD2
0.204	FLOOD3
0.408	FLOOD4
0.816	FLOOD5
1.224	FLOOD6

The user may construct other curves on the diagram and insert corresponding data in a similar element for use in PACKED.

TABLE 1

(Data from Treybal [9] p.168)

Liquid Holdup in Packed Columns:

$$\phi_t = \phi_o + \phi_s \; ; \quad \phi_{tW} = \phi_{oW} + \phi_{sW} \; ; \quad \phi_o = \phi_{oW}{}^{H^*}$$

Correlations: $\phi_{tW} = \dfrac{PHTW(10^{-5})(L')^{BETA}}{d_s{}^2}$

where $BETA = 0.965(d_s)^{0.376}$

and d_s is the diameter of a sphere of same surface area as a single packing particle.

$$\phi_{sW} = \frac{PHSW}{(d_s)^{SWP}}$$

Raschig Rings		Size (mm)			
Text symbol	Program Symbol	12,7	25,4	38,1	50,8
d_s (ft)	DS	0,0582	0,1167	0,174	0,238
	PHTW	2,25			
	PHSW	0,00104			
	SWP	1,21			

Berl Saddles

	12,7	25,4	38,1
DS	0,0532	0,105	0,155
PHTW	2,50		
PHSW	0,00032		
SWP	1,56		

These data are contained in subroutine PAKING. Other factors, and the following correlations, are built into PACKED.

(a) ϕ_s

Raschig rings $$\phi_s = \frac{6,85(10^{-5})\mu_L'^{0,02}\sigma'^{0,99}}{d_s^{1,21}\rho_L^{0,37}}$$

Berl saddles $$\phi_s = \frac{1,641(10^{-4})\mu_L'^{0,04}\sigma'^{0,55}}{d_s^{1,56}\rho_L^{0,37}}$$

(b) H*

Ceramic Raschig rings

$\mu_L' < 12,0$ cp $$H^* = \frac{0,897L'^{0,57}\mu_L'^{0,13}}{\rho_L^{0,84}(0.1183L'^{0,430}-1)}\left(\frac{\sigma'}{73}\right)^{0,925-0,262\text{Log }L'}$$

$\mu_L' > 12,0$ cp $$H = \frac{0,575L'^{0,57}\mu_L'^{0,31}}{\rho_L^{0,84}(0,1183L'^{0,430}-1)}\left(\frac{\sigma'}{73}\right)^{0,925-0,262\text{Log }L'}$$

Ceramic Berl saddles

$\mu_L' < 20,0$ cp $$H = \frac{1,291L'^{0,57}\mu_L'^{0,13}}{\rho_L^{0,84}(0,212L'^{0,413}-1)}\left(\frac{\sigma'}{73}\right)^{1,033-0,262\text{Log }L'}$$

$\mu_L' > 20,0$ cp $$H = \frac{0,752L'^{0,57}\mu_L'^{0,31}}{\rho_L^{0,84}(0,212L'^{0,413}-1)}\left(\frac{\sigma'}{73}\right)^{1,033-0,262\text{Log }L'}$$

TABLE 2

(Data from Treybal [9] p.166)

Shulman Packing-data Tables.

Interfacial area of packing. $a_{AW} = mV'^{n_L'^p}$

Range of L' 2441-7323 kg/hr m^2

Packing	Size	Text Symbol	m	n		p
	mm	Prog.Symbol	AMF1	ANF1A **	ANF2A	PF1
Raschig Rings	12,7		8200	3,15(10^{-4}L')	-0,3	-1,04
	25,4		0,274	0,0	0,0	0,552
	38,1		1,82	0,675(10^{-4}L')	-0,1013	0,274
	50,8		0,401	0,0	0,0	0,481
Berl Saddles	12,7		0,0336	0,0	0,0529	0,761
	25,4		15,89	0,686(10^{-4}L')	-0,1029	0,0
	38,1		0,613		-0,0508	0,455

Range of L' 7323-21969 kg/hr m^2

Packing	Size	Text Symbol	m	n		p
	mm	Prog.Symbol	AMF2	ANF1B	ANF2B	PF2
Raschig Rings	12,7		9,32	0,151(10^{-4}L')	+0,148	-0.111
	25,4		463,	0,528(10^{-4}L')	-0,0793	-0,47
	38,1		4,85	0,148(10^{-4}L')	-0,022	0,140
	50,8		0,95	0,0	0,0	0,362
Berl Saddles	12,7		2,54	0,0	+0,0529	0,170
	25,4***		238,	0,42(10^{-4}L')	-0,063	-0,359
	38,1***		465,	0,325(10^{-4}L')	-0,0996	-0,1355

** Factor (10^{-4}L') shown with values of ANF1A and ANF1B is included in the program. Only the coefficient is required in the data subroutine (see listing of subroutine PAKING).

*** Data for V' < 3905 kg/hr m^2. The values shown above are

included in PAKING. For standard tutorial problems there is no need to change the data if V' exceeds 3905 [9].

TABLE 3

(Data from Treybal [9] p.162)

Packing Characteristics:

Packing*		Nominal size (mm)			
Raschig Rings		12,7	25,4	38,1	50,8
Symbols					
Text	Program				
C_f	CF	640	160	95	65
ε	E	0,63	0,73	0,71	0,74
Berl Saddles		12,7	25,4	38,1	
Text	Program				
C_f	CF	380	110	65	
ε	E	0,63	0,69	0,75	

* Both packings are made of ceramic material

APPENDIX B

NOMENCLATURE

The nomenclature tables list all major FORTRAN text variables with the corresponding standard symbols used in the text. Where possible, text variables comply with the conventions recommended by the AIChE Report [78]. FORTRAN variables not listed below represent intermediate results in the calculation procedure, and are not carried through the program. Several minor subroutine calling list arguments fall into this category (eg. ALPHA in XYALFA is the value of α returned to the calling program).

B.1 NOMENCLATURE CODES

Code	Program
BAT	BATCH
BIN	BINARY
PAC	PACKED
MUL	MULCOM
PLO	PLOTTER
SPL	Special - variables falling in this category are specific to the routine in which they appear.
T	Variable Type: R - real I - integer A - alphanumeric
DIM	Dimension type 1 - single dimension 2 - two dimensional array
COM	Common variable - variable appears in COMMON block.
PARA	Dimension parameters (see below).

Subscripts/Arguments:

Text	Program	Purpose
i	I	component number
j	J	general tray number
k	K	general counter
m	M	tray number in stripping section
n	N	tray number in rectifying section

B.2 PARAMETER definitions

For binary systems certain variables have fixed dimensions depending on their purpose. All other arrays are dimensioned using a parameter.

Parameter	Purpose	Program
(a) Binary programs		
NPAK	Data used for height-of-packing integration.	PACKED
NPBS	Data used for batch distillation integration.	BATCH
NPEN	Enthalpy data.	BINARY,PACKED
NPEQ	Equilibrium data.	BATCH,BINARY,PACKED
NPPS	Ponchon-Savarit diagram data.	BINARY,PACKED
NPTC	Theoretical tray construction on x,y diagram.	BATCH,BINARY
(b) Multicomponent programs		
MC	Components	
MP	Trays	

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
A	$A_{i,j}$	R	Absorption factors defined by Equ.5.18	*			*			2	*	MC,MP
A,B,C	A_i, B_i, C_i	R	Antoine constants	*	*	*				1	*	2
AA,AAW	a, a_{AW}	R	Specific interfacial surface area for mass transfer $ft^2 ft^{-3}$ (Equ.4.12,4.16)			*						
ABCIS	ABCIS	R	x-axis coordinate on flooding correlation chart (Fig.1 Appendix A equ.4.17)			*						
ABS		R	ABCIS as used in subroutine FDATA						*			
ACC		R	Convergence limit for iterative calculations in subroutines RMIN and OPLINE						*			
ACCLIM		R	Limit of accuracy set for convergence of θ-method				*				*	
ACP,BCP,CCP,DCP	a_i, b_i, c_i, d_i	R	Coefficients for Kobe [55] heat capacity correlation (Equ.3.19)		*	*				1	*	2
AKB1 -AKB4	a,b,c,d	R	Coefficients for K_{b_k} correlation (Equ.7.4.9(a))				*				*	
AK1 - AK4	a_i, b_i, c_i, d_i, e_i f_i, g_i as required by user	R	Coefficients for $K_{i,j}$ and $\alpha_{i,k}$ correlations (Equ. 7.4.8,4.9(b),4.10)				*			1	*	MC
AL,AV	L,L',V,V'	R	Liquid and vapour molar flowrates in each section of column for McCabe-Thiele designs		*					1	*	11
ALBKGH		R	Conversion factor: lbs ft^{-2} to kg m^{-2}			*						
ALENX,ALENY		R	Length of x and y axes (inches) in plotting subroutine					*				
ALF		R	Value of α sought during calculations						*			
ALFA	$\alpha_{1,2}$	R	Relative volatilities as defined by equ 2.2 corresponding to equilibrium compositions XALF,YALF	*	*	*				1	*	NPEQ
ALFCON		R	Constant relative volatility value (may be assumed for McCabe-Thiele designs)	*	*						*	
ALIM,BLIM		R	Limits of integration in subroutine SIMPS						*			
ALIQ,AVAP	L_F, V_F	R	Liquid and vapour fraction of feed (kgmoles hr^{-1})				*				*	
ALTI	L'	R	Superficial liquid mass velocity lb hr^{-1} ft^{-2} in packed columns.			*						
ALXT,ALYT		R	Title-positioning data for multicomponent diagrams					*		2	*	4,5
ALX-TF,TF1,TR		R										
TRN,TR1,TR2		R	Adjustment of diagram-title and						*			
ALY-TR,TR1		R	symbol legend positions									

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
ALX1,ALY1		R	Origin of each diagram set relative to point of completion of previous diagram The overall layout is constant although the overall size may change					*			*	
ALX4,ALY4		R	Final positioning of pen on completion of whole job.						*			
AMAT		R	Array required in least squares fit procedure by UNIVAC library routine "GJR" (see subroutine FDATA)						*	2		20,20
AMF1,AMF2	m	R	Coefficient in equ.4.16. Defined in Appendix A Table 2			*						
AMLCON,AMVCON		R	Conversion factors: centipoise and micropoise - lb $hr^{-1}ft^{-1}$.			*						
AMUML,AMUMV	μ_{m_L},μ_{m_V}	R	Liquid and Vapour phase viscosities (Equ. 4.28 and 4.30) returned by subroutine VISCL and VISCV respectively					*				
AN,BN,CN	A_i,B_i,C_i	R	Antoine constants in MULCOM (see equ. 7.4.6)				*			1	*	MC
ANFUNC	n	R	Exponent in equ. 4.16. A function of ANFIA and ANFIB or ANF2A and ANF2B			*						
ANF-1A,1B,2A,2B		R	Factors for computing ANFUNC (see Appendix A Table 2)			*						
AREA		R	Cross-sectional area of packed column			*						
ASFSM		R	Conversion factor: $ft^2ft^{-3}m^2m^{-3}$			*						
ASIZ		R	Size of character (inches) used on plotted diagrams					*			*	
ASTRT,BSTRT		R	Origin of axes for x,y diagram in subroutine PLOTMT						*			
ASURF	a_A	R	Computed value of a_A in m^2m^{-3} at points in the column			*				1		NPAK
ATM		R	Conversion factor: atmosphere pressure to $lb.in^{-2}$	*	*	*	*					
AVEL,AVEV		R	Average molecular weight of liquid and vapour phases.						*			
B	b_i	R	Molal withdrawals rate of component i in Bottoms (equ. 5.16)				*			1	*	MC
BD	b_i/d_i	R	Ratio of b_i and d_i as defined in equ. 5.34				*			1	*	MC
BETA	β	R	Factor defined in Table 1, Appendix A (computed in subroutine PAKING)						*			
BM,DM		R	Factors used in enthalpy balance calculations. See sections 5.53 and 5.562 (from equ. 5.10)				*				*	
BOT,TOP,FEED		R	Limits of integration defining extremities of packed section of column (see Fig. 4.1)			*						
BTOT	B	R	Total molar Bottoms product withdrawal rate (kg moles/hr)	*	*	*	*				*	

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
CB,SLOPB		R	c,m in y=mx+c for operation line in bottom section of column for open steam operation.						*			
CDLDV		R	Molar ratio of liquid and vapour in two phase distillate product.						*			
CF	C_f	R	Characterization factor of packing (Table 3, Appendix A)			*						
CFACT,SLOPE		R	c,m in y=mx+c used generally in calculations.						*			
CFCT,SLPE		R	c,m in y=mx+c for each operating line on McCabe-Thiele diagram.		*					1	*	12
CHEKL,CHEKV		R	Liquid and vapour flowrates from previous iteration.						*			
CHEKLU,CHEKLL		R	Upper and lower limits constraining new liquid flowrates.						*			
CHEKVU,CHEKVL		R	Upper and lower limits constraining new vapour flowrates.						*			
CMAT		R	Array required by UNIVAC routine GJR (see AMAT)						*	1		20
COEFF		R	Solution vector in least squares curve fitting procedure.						*	1		20
COMPNO		R	Storage of component numbers 1-NC in real form for use by PLOTTER.				*		*	1		MC
CONFAC		R	Conversion factor: kJ/hr - kW									
CP	C_{Pi}	R	Heat capacity defined by Kobe [55] correlation (equ.3.19).						*			
CPCONV, CPMW		R	C_p conversion factors (see section 7.235).		*	*					*	
					*	*				1	*	2
CRHO	c	R	Molar density of heavy component (liquid phase) (equ.4.9)			*						
D	d_i	R	Molal withdrawal rate of component i in the distillate (equ.5.13)				*			1	*	MC
DEL		R	General difference factor in calculation						*			
DELB,DELD	Δ_B, Δ_D	R	Enthalpy coordinates of Ponchon-Savarit Δ points.		*			*			*	
DELBC,FACT, NFACT		R I	Factors used in determing the number of integration intervals in the stripping section based on the size of the interval used in the rectifying section (see subroutine OPLINE)						*			
DELDM	Δ_{Dm}	R	Δ point defining minimum reflux in subroutine RMIN (equ.3.25)						*			
DELTAC,DELTAP		R	Pressure drop across condenser and trays respectively (in MULCOM)				*				*	
DIAM,DMI		R	Diameter of packed column (metres)			*						
DIFUSL,DIFUSV	D_L, D_V	R	Calculated liquid and vapour diffusivities.			*						

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
DL,DV		R	Liquid and vapour fractions of a two phase distillate.		*						*	
DLCON,DVCON		R	Conversion factors: cm^2sec^{-1}- ft^2hr^{-1}			*						
DS	d_S	R	Diameter of sphere with same surface area as a packing particle, ft. (see Appendix A,Table 1).			*						
DTOT		R	Total molal Distillate product withdrawal rate.	*	*	*	*				*	
E	ϵ	R	Fractional void volume (dry bed) equ.4.11 (see Appendix A, Table 3).			*						
EFF	E_M	R	Murphree tray efficiency factor for Binary system (equ. 3.7).		*						*	
EFFV	$E_{V_{i,j}}$	R	Holland's vaporization efficiency (equ.5.61),				*			2	*	MC,MP
ENCONV		R	Conversion factor: Btu/lbmole - kJ/kgmole.		*	*	*					
EO	ϵ_o	R	ϵ for irrigated packing (equ.4.11)			*						
ERROR,ERRD		R	Errors in convergence criteria checks (see section 5.58)				*					
EVENS,ODDS		R	Simpson's Rule factors (see section 2.54).						*			
EXAX,EYAX		R	Ponchon-Savarit diagram titles.					*				
EXTEND		R	Variable spacing diagrams for multicomponent systems.					*				
FACTP1,FACTP2		R	Conversion factors: Used to convert specified pressure units to mm Hg						*			
FA1,FA2		R	Factors in calculation of liquid phase mass transfer coefficients (FA1*FA2 = k_L in equ. 4.9)			*						
FEED	FZ_i	R	Moles of component i in feed (subroutine OUTPUT).				*		*	1		MC
FENTH	H_F	R	Enthalpy of feed as specified for feeds at a greater pressure than the column.				*					
FL,FV	F_L,F_V	R	Mass transfer coefficients defined by equ.4.9 and 4.8.			*						
FLIQ,FVAP,FTEMP, FX		R	Arrays corresponding to COMMON array LIQR, VAPR, TRL,XROP or LIQS, VAPS, TSL,XSOP when FFIND is called by STEPD.						*	1		NPPS
FLOL,FLOV	L_j,V_j	R	Liquid and Vapour flows leaving each tray as calculated in FFIND for Ponchon-Savarit designs.						*			
FLOLIM		R	Factor defining values of CHEKLU, LL etc. in FLOWNE (depends on ICOUNT.)						*			

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
FLOW		R	Array storing feed and side stream flowrates in positional order as determined in MULTFS.		*					1	*	10
FTOT	F	R	Total feed rate for simple columns (kg moles hr^{-1})	*	*	*	*				*	
FTOTS,STOTS		R	Total feed and side stream rates for complex columns.		*					1	*	5
FUNC		R	Values of function to be integrated by Simpson's Rule.	*						1		NPBS
FUNCR,FUNCS		R	Values of function in rectifying and stripping sections in PACKED, to be integrated.			*				1		NPAK
F_1,F_2,F_3,F_4		A	Plotting data file names.				*	*				
-	$f_{i,j}$		Molar feed (or withdrawal if negative) of component i to general stage j (equ.1.2).									
G	θ_T,G	R	Batch time (θ_T)or Boil-up rate (G) as specified. Related by equ.2.15. Unknown value determined from equ.2.14, 2.17 or 2.20.	*								
GC	g_c	R	Gravitational force constant $4,18.10^8$lb.ft.$lb_j^{-1}hr^{-2}$.			*			*			
GOP		R	Moles of open steam required to replace reboiled vapour.		*				*			
H		R	General enthalpy value used in calling lists in HTSOLN and HFIND.						*			
H		R	Size of integration interval used in BATCH and OPLINE (see also NINT).						*			
HB	h_B	R	Total enthalpy of Bottoms product.		*						*	
HD	H_D	R	Total enthalpy of Distillate product regardless of state.		*		*				*	
HEAD1-HEAD4		A	Diagram headings in PLOTTER.					*		1		5
HENTH,XYH		R	Arrays corresponding to HLIQ,XH or HVAP,YH in HFIND.						*	1		NPEN
HEVABS		R	Factor associated with heat of solution data. Equals +1 or -1 for endothermic or exothermic mixing respectively.		*						*	
HF,HFEED	H_F	R	Calculated feed enthalpy kJ $kgmole^{-1}$		*		*				*	
HFUNC	H*	R	Empirical factor in equation 4.15. Defined in Table 1, Appendix A.			*						
HLD,HVD		R	Liquid and vapour phase distillate enthalpies.		*	*			*			
HLF,HVF		R	Liquid and vapour phase feed enthalpies.		*	*			*			
HLIQ,HVAP	h_j,H_j	R	Enthalpy of liquid and vapour stream in columns. kJ.$kgmole^{-1}$ in binary and multicomponent programs.			*				1	*	NPEN
								*		1	*	MP

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
HL1,HL2,HL3 HV1,HV2,HV3		R	Coefficients for enthalpy functions considered in ENERGY (see section 7.432)				*			1	*	MC
HS		R	Calculated heat of solution included in calculation of h_j (see section 3.59)		*							
HSCONV,		R	Conversion factors (see section 7.235)		*						*	
HSMW		R								1	*	2
HSOLN,XHS		R	Heat of solution composition data supplied.		*					1	*	30
HTEST,HVCM		R	Enthalpy factors used in calculations in RMIN HTEST ≠ HF within limit ACC after 500 iterations causes termination of search						*			
I	i	I	Subscript and 'DO' loop counter referring to components.						*			
IBUF		I	Array required by CALCOMP plotting software.					*		1		1000
ICOUNT		I	Counter determining number of overall iterations.				*				*	
INTA,B		I	Numerical definition of : and / in CALCOMP software					*				
INTEQ,I		I	Numerical codes for symbols plotted on diagrams.					*				
IT			Tray number counter in BUBCAL and DEWCAL						*			
IUSE			Counter defining run number for PLOTTER.				*	*	*			
J	j	I	Subscript and 'DO' loop counter referring to tray number.						*			
JC		I	Array required by UNIVAC routine GJR. It is used to return the following data if anderror occurs in GJR: (a) JC(1)=-I where I = last correctly reduced row. (b) JC(1)= I where I = last row before singularity was detected.						*	1		20
K	$K_{i,j}$	R	Equilibrium constants defined by equ.5.1				*			2	*	MC,MP
KF	$K_{i,F}$	R	Equilibrium constants at feed stream				*			1	*	MC
KX,KY	k_x,k_y	R	Mass transfer coefficients, liquid and Vapour phase.			*				1		NPAK
LV	L_j,V_j	R	Liquid and Vapour molar flowrates at points through the packing (kg moles/hr m^2).			*				1		NPAK
LB,LD	$l_{i,m/b_i}$, $l_{i,n/d_i}$	R	Defined by equ.5.29 and 5.25				*			2	*	MC,MP
LFREQ,LFREQ1		I	Code defining plotting procedure in CALCOMP software.					*				
LH	$LH_{i,T_{BP}}$	R	Latent heat values for each component (binary).		*					1	*	2

LHCONV, LHMW		R R	Conversion factors: see section 7.235							1	*	2
LHT	$LH_{i,T}$	R	Latent heat of component i calculated by equ 3.20.						*			
LIQR,LIQS	L_n, L_m	R	Liquid molar flowrates calculated at points (XROP,YROP) and (XSOP,YSOP) in the rectifying and stripping sections.		*					1	*	NPPS
LMFR,VMFR	$l_{i,j}, v_{i,j}$	R	Molal flowrate of component i leaving tray j in liquid and vapour phase respectively.				*			2	*	MC,MP
LPROF	L_j	R	Molar liquid profile through multi-component column.				*			1	*	MP
LRTOT,LSTOT	L_n, L_m	R	Molar liquid flowrates in rectifying and stripping sections.				*			1	*	MP
LRWT,LSWT		R	Liquid mass flowrates corresponding toLIQR,S in binary system. Required by PACKED.		*	*				1	*	NPPS
M	m	I	Subscript and 'DO' loop counter referring to trays in the stripping section of a column.						*			
MALF		I	Counter indicating constant relative volatility (=2) or otherwise (=1). Also used as termination counter (= 1000).	*	*	*						
MC,MP			See PARAMETER definitions : Section B.2									
MCAN		I	Counter activation (= 1) or suppressing (= 2) the conversational facility.	*	*	*	*					
MEFF		I	Counter indicating type of efficiency data supplied.				*				*	
MFIN		I	(See NFIN)									
MMD		I	Counter causes efficiency factors to be ignore for total condenser or feed stream bubble point and dew point calculations (see section 5.55).				*					
MN,NCALL		I	Counters directing iterative search procedure (see subroutine SEARCH section 2.55).						*			
MPTS		I	See NPTS									
MPT1		I	MPTS + NFIN (defines bottom of rectifying section relative to NTOT).									
MSTOP		I	Terminates stepping off of McCabe-Thiele stages in XDSTEP.	*								
MT		I	Total number of theoretical stages stepped off on x,y diagram in STEPD,		*							
MTIMES		I	Counter prevents recalculation of tray pressure profile and activates conversational facility for adjusting variables to be used in second and subsequent runs.				*					

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
MW		R	Molecular weight of components in binary systems.		*	*				1	*	2
N	n	I	Subscript and 'DO' loop counter referring to the rectifying section of a column.						*			
NBCP,NCCP,NDCP		I	Exponents relating to Kobe function coefficients (see section 7.234)						*			
NBDC		I	Counter selecting bubble point and dew point calculation procedure: = 1 for Antoine constants = 2 for Equilibrium data	*	*							
NC	NC	I	Number of components in multi-component systems.				*				*	
NCERR,NLOOP		I	Counters used in OPLINE to ensure that the number of points chosen does not exceed the dimension limit NPPSC = NPPS - 2						*			
NCODE		I	Counter selects descriptive output in TEACH.						*			
NCOND		I	Counter = 1 - total condenser operation = 2 - partial condenser operation.	*	*	*	*				*	
NCOUNT,MCOUNT		I	Equivalent to NPTS,MPTS in mainline routines. Define number of points on rectifying and stripping section operating lines. (see section 3.512)						*			
NCR		I	READ key (= 8 on UNIVAC system)	*	*	*	*	*			*	
NCURVE		I	Counter selecting correlation curve data for FDATA from Fig 1,Appendix A (Equivalent to NFLOOD from PACKED)			*			*			
NDAT		I	Number of equilibrium data points in XEQUIL. YEQUIL (= NDATA + 2).						*			
NDATA		I	Number of equilibrium data points excluding 0.,0. and 1.,1.	*	*	*					*	
NEN		I	Counter defining enthalpy calculation (= 1 or 2 from QFLASH,= 3 from MULCOM)						*			
NENTH		I	Counter indicating method of performing enthalpy calculations, see section 7.432 (= 0 if enthalpy is not considered).				*				*	
NENTH		I	Number of enthalpy-composition data points defining lines on Ponchon-Savarit diagram.		*						*	
NEQ		I	Equivalent to NEN for equilibrium calculations (= 1 from QFLASH, = 3 from MULCOM).						*			
NEQUIL		I	Counter indicating equilibrium constant correlation used.				*				*	

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
NF		I	Feed tray position specified by user or Number of feed streams specified for McCabe-Thiele design.		*		*				* *	
NFACT		I	See DELBC.									
NFIN,MFIN		I	Point on operating line representing the bottom of the column packing. Defined by stepping off the reboiler stage.			*			*			
NFLOOD		I	Counter indicating selected curve on on Fig. 1, Appendix A.			*			*			
NF1,NF2		I	Feed tray position and number of tray above feed tray. (NF1 = NF + 1 in internal coding in MULCOM section 8.2)				*				*	
NG1,NG2,NG3,NG4		I	Write keys for plotting data files. Defined in DATA statements in each mainline.	*	*		*				*	
NHCAT,NLCAT		I	Counters indicating component category (see section 7.32)			*						
NHS		I	Number of HSOLN,XHS data pairs supplied.		*						*	
NINT		I	Number of intervals of size H chosen for numerical integration procedure in BATCH and OPLINE.	*	*	*					*	
NJ		I	Counter identifying base components when equilibrium data is supplied in terms of relative volatilities. (section 5.55) Also used to terminate the run if errors are detected in subroutines.				*				*	
NLP		I	WRITE key for data file setup for for detailed result table.				*				*	
NN		I	Number of operating lines on McCabe-Thiele x,y diagram (= NF + NS + 1).		*				*		*	
NN1		I	Number of points defining the operating lines (= NN + 1).		*				*			
NOP		I	Counter: = 1 for McCabe-Thiele diagrams. = 2 for Packed column designs = 3 for Ponchon-Savarit tray tray column designs.		*	*					*	
NOP1		I	Counter defining Batch operation: = 1 Varying reflux = 2 Constant reflux	*								
NORD		I	Order of polynomial with best fit (subroutine FDATA).						*			
NOS		I	Counter: = 0 - partial reboiler = 1 - open steam									
NOUT		I	Counter limiting number of detailed result tables to 5.				*					

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
NOWN		I	Counter indicating enthalpy calculation routine used.				*				*	
NP	N	I	Number of trays in the column specified by the user. Excludes reboiler and condenser.				*				*	
NPACK		I	Counter indicating specified size of packing.			*						
NPH,NPHYS		I	Counters ensure that physical property data are read as required.			*						
NPOINT		I	Indicates point of maximum liquid flowrate in a packed column.			*						
NPOLY		I	Equals (NORD + 1) in FDATA (number of terms in polynomial).						*			
NPTS,MPTS		I	Number of points defined on rectifying and stripping section operating lines.		*	*						
NP1,NP2		I	Top tray and condenser numbers (interval numbering system. See section 8.2).				*				*	
NQVAL		I	Counter indicating feed conditions = 1 - feed at feed tray pressure = 2 - feed at pressure > feed tray pressure.				*				*	
NROM,NR		I	Counters indicating availability of solution density - composition data, and when this is to be used.									
NRFIN		I	Point on rectifying section operating line representing the bottom of this section of packing defined in NFIN.			*						
NS		I	Number of side streams in complex columns.		*						*	
NSORT		I	Counter indicating the phase controlling the mass transfer.			*						
NSPEC		I	Counter: = 1 if $X_{D_{AVE}}$ is not specified = 2 if $X_{D_{AVE}}$ is specified.	*								
NTC		I	Number of points defining stepped stages on McCabe-Thiele x,y diagram.	*	*							
NTIME		I	Counter causes subroutine ENTHAL to read data once only.						*			
NTIMES,NTIM		I	Counters in plotting program defining number of curves plotted.					*				
NTOT		I	Total number of points defining Ponchon-Savarit operating lines = NPTS + MPTS.		*	*						
NTP		I	General WRITE key for writing to terminal in demand mode or line printer in batch mode (= 5 on UNIVAC system).	*	*	*	*	*			*	

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
NTRA		I	Indicates position of feed tray in data tables for Ponchon-Savarit design.		*							
NTRAY		I	Stores feed and side stream tray numbers sequentially during stepping procedure.		*					1	*	12
NTYPE		I	Counter indicating type of packing specified.			*						
ORD		R	Ordinate value of point on flooding correlation chart (see Fig 1, Appendix A).			*						
PAL,P1	P_i	R	Vapour pressure of light component (equ.2.9).	*	*				*			
PCOND	P_T	R	Condenser pressure in mm Hg.				*				*	
PCONDA	P_T	R	Specified condenser pressure in convenient units.	*	*	*	*					
PFACT1,PFACT2		R	Factors in empirical expression defining ϕ_s (see Table 1, Appendix A.)			*						
PF1,PF2	1	R	Empirical constants shown in Table 2, Appendix A (equ.4.16).			*						
PHIO	ϕ_o	R	Operating holdup ft^3 liquid/ft^3 packed volume (equ.4.15).			*						
PHIOW	ϕ_{ow}	R	Operating holdup with respect to pure water.									
PHIS	ϕ_s	R	Static holdup (see Table 1, Appendix A).			*			*			
PHISW	ϕ_{sw}	R	Static holdup with respect to water.			*			*			
PHIT	ϕ_t	R	Total holdup (equ.4.13).			*			*			
PHITW	ϕ_{tw}	R	Total holdup with respect to water (equ. 4.14).			*			*			
PHSW,PHTW		R	Empirical factors shown in Table 1, Appendix A.			*			*			
POWER		R	Exponent relating to multicomponent enthalpy correlation (see section 7.43)				*		*			
PRESS,PRES		R	Tray pressure profile in mm Hg and kPa respectively.				*			1	*	MP
PT,PT1	P_T	R	System pressure in mm Hg and kPa respectively (PT = PCONDA converted to mm Hg).	*	*	*					*	
PVAP	$P_{i,j}$	R	Multicomponent vapour pressure data (equ.2.9) used in subroutine EQUCON.				*		*			
-	q'_j		Heat loss from general stage j (equ.1.3)									
Q	q	R	Feed condition for simple columns.		*	*	*				*	
QB	Q_B	R	Reboiler duty (kW) (Binary systems)		*	*					*	
QC	Q_C	R	Condenser duty (kW)		*	*	*				*	
QF,QFTOT	H_F,F,H_F	R	Molar enthalpy and heat content of multicomponent feed (kJ $kgmole^{-1}$;kW)				*					

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
QF,QS	q_F,q_S	R	Condition of feed and side streams in complex columns.		*					1	*	5
QFS	q_F and q_S	R	Array storing q_F and q_S values as arranged in MULFS.		*					1	*	10
QMIX,ZMIX		R	q value and composition of "feed" representing all feed and side streams in an overall mass balance for complex columns.		*							
QR		R	Reboiler duty for multicomponent systems (kW).				*				*	
QX,QY		R	Arrays storing coordinates of NN1 points defining operating lines in binary McCabe-Thiele systems.		*					1	*	12
R	R	R	Binary system reflux ratio	*	*	*					*	
RHOCON		R	Conversion factor for density data: g mℓ → lb ft^{-3}.			*			*			
RHOL,RHOL1,2		R	Liquid mixture and component densities from RHOLIQ.			*						
RHOV		R	Calculated vapour mixture density									
RHOW		R	Density of water from RHOH2O									
RM	R_M	R	Minimum reflux ratio in binary systems		*	*					*	
RR	R	R	Reflux ratio in multicomponent systems				*				*	
S	$S_{i,j}$	R	Stripping factors (equ.5.22)				*			2	*	MC,MP
SCHL,SCHV	Sc_L,Sc_V	R	Liquid and vapour phase Schmidt numbers			*			*			
SIGCON		R	Conversion factor for surface tension data.			*			*			
SIGMA,SIG1,SIG2	σ	R	Mixture and component surface tension data from SURTEN (see equations 4.22 - 24).			*			*			
SLIQ,SVAP	$\Sigma l_{i,j}$ $\Sigma v_{i,j}$	R	Summations according to equations 5.43 and 5.44.				*			1	*	MP
SLPE,CFCT		R	Slope and intercept (y = mx + c) of McCabe-Thiele operating lines.		*	*				1	*	12
STR,STR1		A	Names used in printing result table by BINRES.		*							
SUM,SUMINT		R	Result of numerical integration from SIMPS.	*		*			*			
SUMB,SUMD	$\Sigma b_{i_{co}}$ $\Sigma d_{i_{co}}$	R	Summation of correlated values in BDCORR.				*					
SUMD1	$D_{CALC} = \Sigma d_{i_{co}}$	R	Value from BDCALC used in convergence test ($D_{CALC} - D_{SPEC}$).				*					
SUMERR		R	Sum of squared errors in least squares fit (FDATA).			*						

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
SWP		R	Exponent in expression for ϕ_s (see Table 1, Appendix A).			*			*			
SX,Y,XF,YF	$\Sigma x_{i,j}, \Sigma y_{i,j}$ $\Sigma x_{i,F}, \Sigma y_{i,F}$	R	Summation of composition values in OUTPUT shown in detailed result table.				*					
S1,S2,S3,S4		R	Enthalpy data returned to QFLASH for use in flash calculations.				*		*			
T,T1	T_j	R	$T = T_j * 10^{-2}$ and T1 is T_j converted to units required by equilibrium or enthalpy correlations.				*			1	*	NPEQ
TB,TBPT,TBUB,TBUBPT		R	General bubble point values.						*			
TBASE	T_b	R	Base temperature for enthalpy calculations.		*						*	
TBOT,TTOP		R	Estimate of reboiler and condenser temperatures used to obtain initial temperature profile for Thiele Geddes method.				*				*	
TBP	T_{BP}	R	Normal boiling points of light and heavy components.		*					1	*	2
TC	T_c	R	Critical temperature of each component.		*					1	*	2
TCONV		R	Conversion factor (temperature): $^oC \rightarrow K$.									
TD,TDPT,TDEW, TDEWPT		R	General dew point values						*			
TEMP		R	Temperature profiles in ^oC (MULCOM)				*			1	*	MP
TEMPF		R	Temperature profiles in ^oF (subroutine RUNRES).				*			1	*	MP
TF,TFEED	T_F	R	Feed temperatures for Binary and Multicomponent systems respectively (^oC).		*		*				*	
TG		R	General temperature value used to start iterative calculation procedures.		*						*	
THETA	θ	R	Computed value of θ (equ.5.50).				*				*	
THEOLD		R	Starting or previous value of θ.				*					
TIME	θ_T	R	Either batchtime or boil-up rate (see G).	*								
TITLES		A	Names of components in all distillation programs.	*	*		*			2	*	2,5 or MC,5
TITL,TITLT		A	Titles used in plotting routine.					*				
TNEW,TOLD		R	Forcing procedure limiting change profile in BUBCAL and DEWCAL.				*		*			
TRANK		R	Conversion factor in ENERGY: $^oF \rightarrow {}^oR$.				*		*			
TRAYNO		R	Total number of trays, including fractional part of any, stepped off on x,y diagram.		*						*	

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
TRAYNO		R	Tray numbers stored as real variables for use in plotting routines (CALCOMP software will not accept integers - see COMPNO).				*			1		MP
TRL,TRV		R	Temperatures of streams corresponding		*	*				1	*	NPPS
TSL,TSV		R	to flowrates LIQR,VAPR,LIQS,VAPS. Used in PACKED.		*	*				1	*	NPPS
T2B,T2D		R	Bubble point and dew point of the liquid and vapour mixtures for which HL and HV are required.		*				*			
U1,U2,U3,U4		R	Used to assign WRITE keys NG1 - NG4 to files F1 - F4 in READIN.				*		*			
V		R	Array required by UNIVAC routine GJR specifying the type of operation required. In this application V(1)=4 (line 54) indicates that the equations in AMAT are to be solved by GJR.			*			*	1		2
VAP	V	R	Total quantity of vapour boiled up during batch cycle (equ. 2.15)	*								
VB,VD	$v_{i,m/b_i}$ $v_{i,n/d_i}$	R	Ratios defined by equation 5.28 and 5.26.				*			2	*	MC,MP
VLOAD	V'	R	Superficial vapour mass velocity kg mm^2 at NPOINT.			*						
VMOLE	V	R	Superficial vapour molar velocity kgmole hr m^2 at NPOINT.			*						
VPDL		R	Specified vapour pressure-drop through packing. (FDATA fits polynomial to corresponding curve Fig. 1, Appendix A).			*						
VPROF		R	Molar vapour profile through multi-component column.				*			1	*	MP
VRTOT,VSTOT		R	Molar vapour flowrates in rectifying and stripping sections respectively.				*			1	*	MP
VYR,VYS	V.y or L.x	R	Data stored depends on form of equation 4.3 used for integration. V.y - for vapour phase limiting mass transfer L.x - for liquid phase limiting mass transfer.			*				1		NPAK
W	W	R	Residue liquid in batch operation	*								
X,Y		R	Coordinates of points defining stepped stages on x,y diagrams.	*	*					1	*	NPTC
X,Y		R	Compositions of liquid and vapour streams leaving each tray (related by equ.5.1).				*			2	*	MC,MP
XALF,YALF		R	Equilibrium composition data defining equilibrium curve on x,y diagram.	*	*					1	*	NPEQ
XB,YB	x_B,y_B	R	Coordinates of point defining lower end of the bottom operating line		*						*	

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
XC,YC	x_i, y_i	R	Point of intersection of top operating line and 45° diagonal on x,y diagram (see Fig. 3.2).		*						*	
XCR,YCR	x	R	Point of intersection of Ponchon-Savarit operating lines (see Fig 3.3)		*	*					*	
XD,YD	x_D, y_D	R	Coordinates of point representing distillate product on x,y diagram see (Fig. 3.2).		*						*	
XDATA,YDATA		R	Values used in numerical integration in BATCH.	*						1		NPBT
XDAVE	$x_{D_{AVE}}$	R	Calculated average distillate composition.	*								
XEQUIL,YEQUIL		R	Copy of XALF,YALF, but includes 0.,0. and 1.,1. for result table and plotting purposes.	*	*					1	*	NPEQ
XF,YF	x_F, y_F	R	Composition of liquid and vapour phase of feed.				*			1	*	MC
XFLO,YFLO	$L_F x_F$ $V_F y_F$	R	Moles of each component in liquid and vapour phase of feed.				*		*			
XH,YH		R	Compositions corresponding to HLIQ and HVAP.		*	*				1	*	NPEN
XI,YI	x_i, y_i	R	Equilibrium compositions at the liquid-vapour phase interface (see Fig 4.1)			*				1		NPAK
XNTC		R	Computed value of residue composition used in iterative determination of XDATA,YDATA.	*								
XOUT		R	Composition value used in calling list to PSOPXY.		*							
XQM,YQM		R	Point of intersection of overall operating lines in complex columns. (see QMIX and ZMIX)		*				*			
XROP,YROP		R	Coordinates of points defining		*					1	*	NPPS
XSOP,YSOP			Ponchon-Savarit operating lines on x,y diagram.		*					1	*	NPPS
XSPEC		R	Specified average distillate composition (see NSPEC).	*								
XW		R	Residue composition at end of batch run.	*								
YINCPT		R	Y-axis intercept (c in y=mx+c) of operating line on batch x,y diagram.	*				*				
Z	z_F	R	Overall composition of feed in binary systems.	*	*	*					*	
Z	$z_{i,F}$	R	Overall composition of feed in multicomponent systems.				*			1	*	MC
ZF,ZS		R	Overall composition of feed and side streams - complex columns. Arrays ZF1, ZS1 required in MULTFS.		*					1	*	5

PROGRAM VARIABLE	TEXT SYMBOL	T	DESCRIPTION	BAT	BIN	PAC	MUL	PLO	SPL	DIM	COM	PARA
ZFS		R	Combined ZF and ZS as arranged in order by MULTFS (see QFS).		*					1	*	10
ZRECT, ZSTRIP	Z_R, Z_S	R	Height of packing required in rectifying and stripping sections respectively.			*			*			
ZFLO, ZFS		R	Flowrates and compositions of feed and side streams for plotting purposes.					*		1		15

APPENDIX C

The FORTRAN listings of all program elements contained in the package are shown here. The name of the mainline or subroutine is displayed on the bottom right hand corner of each page. Line numbers referred to in the main text are shown to the right of the FORTRAN statement.

Programs are listed in the order discussed in the main text.

Subroutine TEACH: This subroutine provides a brief description of each program in the package, excluding PLOTTER (see User Manual, Chapter 7). It is self-explanatory and is not discussed in the text.

Chapter 2: Program BATCH

Mainline BATCH (section 2.341 of main text)

Subroutines associated with BATCH:

Subroutine	section
XDSTEP	2.51
BINBPT	2.521
BINDPT	2.522
XYALFA	2.53
SIMPS	2.54
SEARCH	2.55

Chapter 3: Program BINARY

Common block BINCOM

Mainline BINARY (section 3.34 of main text)

Subroutines associated with BINARY:

Subroutine	section
READBD	3.51
CONDEN	3.52
FLASHB	3.53
EQUIL	3.54
MULTFS	3.55
SIMPLE	3.56
PONSAV	3.57
HTCAP	3.58

Subroutine	section
HTSOLN	3.58
HFIND	3.510
RMIN	3.511
OPLINE	3.512
FLOWS	3.513
AVEMW	3.514
CHEC	3.515
STEPD	3.516
PSOPXY	3.517
FFIND	3.518
BINRES	3,519

Chapter 4: Program PACKED

Common block PACCOM

Mainline PACKED (section 4.34 of main text)

Subroutines associated with PACKED:

Subroutine	section
PACRES	4.51
PACTYP	4.52
PAKING	4.53
FDATA	4.54

Physical property subroutines:

RHOLIQ	4.551
SURTEN	4.552
DIFLIQ	4.553
DIFVAP	4.554
VISCL	4.555
VISCV	4.556
RHOH2O	4.557

Chapter 5: Program MULCOM

Common block MULTICOM

Mainline MULCOM (section 5.36 of main text)

Subroutines associated with MULCOM:

Subroutine	section
READIN	5.51
EQUCON	5.52
ENERGY	5.53
QFLASH	5.54
BUBCAL,DEWCAL	5.55
FLOWEQ,FLOWNE	5.56
ABSTRP,VBLB,	5.57
VDLD,BDCALC,	"
BDCORR,VLRECT,	"
VLSTRP	"
THECAL	5.58
RUNRES,OUTPUT	5.59

Chapter 6: Program PLOTTER

Mainline PLOTTER (section 6.2 of main text)

Subroutines associated with PLOTTER:

Subroutine	section
PLOTTS	6.31
PLOTFS	6.32
PLOTDS	6.33
PLOTCS	6.34
PLOTBT	6.35
PLOTMT	6.36

```
      SUBROUTINE TEACH(I)                                                0001
      NTP=5                                                              0002
      GO TO (100,200,300,400),I                                          0003
  100 CONTINUE                                                           0004
      WRITE(NTP,1000)                                                    0005
 1000 FORMAT(/T2,'THIS IS A BINARY BATCH DISTILLATION SIMULATION PROGRAM  0006
     1,',/T2,'SYSTEMS OPERATING UNDER EITHER VARIABLE OR CONSTANT REFLUX  0007
     2',/T2,'CONDITIONS CAN BE CONSIDERED.',/T2,'THE PROGRAM IS LIMITED   0008
     3TO SYSTEMS FOR WHICH NORMAL',/T2,'MCCABE-THIELE ASSUMPTIONS HOLD.'  0009
     4,/T2,'PLOTTING ROUTINES ARE AVAILABLE FOR CONSTRUCTING MCCABE-THIE  0010
     5LE',/T2,'X-Y DIAGRAMS FOR EITHER OPERATION MODE.')                  0011
      WRITE(NTP,1001)                                                    0012
 1001 FORMAT(/T2,'REFER TO THE DISTILLATION PACKAGE MANUAL,SECTION 1,',/  0013
     1T2,'FOR FURTHER INFORMATION.')                                      0014
      GO TO 999                                                          0015
  200 CONTINUE                                                           0016
      WRITE(NTP,2000)                                                    0017
 2000 FORMAT(/T2,'THIS IS A BINARY CONTINUOUS DISTILLATION DESIGN PROGRA  0018
     1M.',/T2,'IT WILL COMPUTE THE NUMBER OF THEORETICAL STAGES REQUIRED  0019
     2',/T2,'TO ACHIEVE A SPECIFIED TOP PRODUCT PURITY.',/T2,'A MURPHREE  0020
     3 PLATE EFFICIENCY FACTOR MAY BE SPECIFIED.',/T2,'THE DIMENSIONS OF  0021
     4 THE COLUMN REQUIRED ARE NOT CALCULATED.'/T2,'SYSTEMS FOR WHICH EI  0022
     5THER MCCABE-THIELE OR PONCHON-SAVARIT',/T2,'ASSUMPTIONS HOLD,MAY B  0023
     6E CONSIDERED.',/T2,'PLOTTING ROUTINES ARE AVAILABLE FOR CONSTRUCTI  0024
     7NG STANDARD',/T2,'DIAGRAMS FOR EITHER DESIGN TECHNIQUE.')           0025
      WRITE(NTP,2001)                                                    0026
 2001 FORMAT(/T2,'REFER TO THE DISTILLATION PACKAGE MANUAL,SECTION 2,',/  0027
     1T2,'FOR FURTHER INFORMATION.')                                      0028
      GO TO 999                                                          0029
```

```
  300 CONTINUE                                                          0030
      WRITE(NTP,3000)                                                   0031
 3000 FORMAT(/T2,'THIS PROGRAM DESIGNS PACKED COLUMNS FOR DISTILLATION O 0032
     1F',/T2,'BINARY SOLUTIONS.',/T2,'THE PONCHON-SAVARIT METHOD IS USED 0033
     2 TO DETERMINE THE OPERATING',/T2,'DIAGRAM.',/T2,'THE HEIGHT OF PAC 0034
     3KING REQUIRED IN EACH SECTION OF THE COLUMN',/T2,'IS DETERMINED BY 0035
     4 INTEGRATION,USING SIMPSONS RULE.',/T2,'MASS TRANSFER COEFFICIENTS 0036
     5 ARE COMPUTED USING SHULMANS',/T2,'CORRELATIONS AND PACKING DATA,' 0037
     6,/T2,'DATA FOR CERAMIC RASCHIG RINGS (4 SIZES) AND BERL SADDLES'  0038
     7,/T2,'(3 SIZES) ARE INCLUDED IN THE PACKAGE.')                    0039
      WRITE(NTP,3001)                                                   0040
 3001 FORMAT(/T2,'REFER TO THE DISTILLATION PACKAGE MANUAL,SECTION 3.',/ 0041
     1T2,'FOR FURTHER INFORMATION.')                                    0042
      GO TO 999                                                         0043
  400 CONTINUE                                                          0044
      WRITE(NTP,4000)                                                   0045
 4000 FORMAT(/T2,'THIS IS A MULTICOMPONENT DISTILLATION SIMULATION PROGR 0046
     1AM.',/T2,'THE CALCULATION PROCEDURE USED IS THAT FIRST PROPOSED BY 0047
     2',/T2,'THIELE AND GEDDES,COMBINED WITH HOLLANDS THETA METHOD OF',  0048
     3/T2,'CONVERGENCE.',/T2,'THE PROGRAM IS LIMITED TO SYSTEMS WITH ONE 0049
     4 FEED AND',/T2,'NO SIDE STREAMS.')                                0050
      WRITE(NTP,4001)                                                   0051
 4001 FORMAT(/T2,'REFER TO THE DISTILLATION PACKAGE MANUAL,SECTION 4.',/ 0052
     1T2,'FOR FURTHER INFORMATION.')                                    0053
  999 CONTINUE                                                          0054
      RETURN                                                            0055
      END                                                               0056
```

SUBROUTINE TEACH

```
C     MAINLINE PROGRAM FOR RATING OF BINARY BATCH DISTILLATION SYSTEMS.
C     EITHER CONSTANT OR VARIABLE REFLUX OPERATIONS CAN BE CONSIDERED.
      PARAMETER NPEQ=55                                                  0001
      PARAMETER NPTC=55                                                  0002
      PARAMETER NPBS=101                                                 0003
      DIMENSION A(2),B(2),C(2),FUNC(NPBS),TITLES(2,5)                    0004
      DIMENSION XDATA(NPBS),XEQUIL(NPEQ),YDATA(NPBS),YEQUIL(NPEQ)        0005
      COMMON ALFA(NPEQ),ALFCON,R                                         0006
      COMMON MALF,NDATA,NP1,NTC                                          0007
      COMMON X(NPTC),XALF(NPEQ),XD,XNTC,Y(NPTC),YALF(NPEQ)               0008
    1 FORMAT( )                                                          0009
    2 FORMAT(A3)                                                         0010
    3 FORMAT(5A6)                                                        0011
 8001 FORMAT(2(2X,F6.4))                                                 0012
 8002 FORMAT(I5)                                                         0013
      DATA MALF,NCAN,NCODE,NINT,NSPEC/1,1,1,100,1/                       0014
      DATA NCR,NG1,NTP/8,11,5/                                           0015
      DATA ATM,FACTP1,FACTP2/14.696,51.714752,7.50061683/                0016
C     START OF PROGRAM ***************************************************
      WRITE(NTP,5101)                                                    0017
 5101 FORMAT(/T2,'ANSWER ALL QUESTIONS YES OR NO.',/T2,'ALL INPUT IN FRE 0018
     1E FORMAT.',//T2,'DO YOU KNOW HOW TO USE THIS PROGRAM?')            0019
      READ(NCR,2) ANAME                                                  0020
      IF(ANAME.EQ.'YES') GO TO 511                                       0021
      CALL TEACH(NCODE)                                                  0022
  511 CONTINUE                                                           0023
      WRITE(NTP,5102)                                                    0024
 5102 FORMAT(/T2,'MODES OF OPERATION: (1) VARIABLE REFLUX',/T22,         0025
     1'(2) CONSTANT REFLUX',/T2,'ENTER 1 OR 2 FOR REQUIRED PROCEDURE.')  0026
      READ(NCR,1) NOP1                                                   0027
      WRITE(NTP,5103)                                                    0028
 5103 FORMAT(/T2,'DO YOU WANT TO ADD A DATA ELEMENT?')                   0029
      READ(NCR,2) ANAME                                                  0030
      IF(ANAME.EQ.'NO') GO TO 512                                        0031
      MCAN=2                                                             0032
      WRITE(NTP,5104)                                                    0033
 5104 FORMAT(/T2,'ADD YOUR DATA ELEMENT.')                               0034
```

```
  512 CONTINUE                                                          0035
      NINTC=NINT+1                                                      0036
      IF(NINTC.GT.NPBS) GO TO 903                                       0037
C     START OF DATA INPUT ***************************************************
      IF(MCAN.EQ.1) WRITE(NTP,5000)                                     0038
 5000 FORMAT(/T2,'ENTER NAMES OF COMPONENTS,,,1 PER LINE,MAX 30 CHARACTE 0039
     1RS EACH.')                                                        0040
      READ(NCR,3) ((TITLES(I,J),J=1,5),I=1,2)                           0041
      IF(MCAN.EQ.1) WRITE(NTP,5001)                                     0042
 5001 FORMAT(T4,'ACCEPTED PRESSURE UNITS: 1=MMHG,2=PSIG,3=PSIA,4=KPA,5=A 0043
     1TM.',/T2,'ENTER CONDENSER PRESSURE AND UNIT CODE, (EG. 760.0,1)')  0044
      READ(NCR,1) PCONDA,NUNITP                                         0045
      IF(NUNITP.EQ.1) PT=PCONDA                                         0046
      IF(NUNITP.EQ.2) PT=(PCONDA+ATM)*FACTP1                            0047
      IF(NUNITP.EQ.3) PT=PCONDA*FACTP1                                  0048
      IF(NUNITP.EQ.4) PT=PCONDA*FACTP2                                  0049
      IF(NUNITP.EQ.5) PT=PCONDA*ATM*FACTP1                              0050
      IF(PT.GT.7600.) GO TO 902                                         0051
      IF(MCAN.EQ.1) WRITE(NTP,5002)                                     0052
 5002 FORMAT(/T2,'WILL ANTOINE CONSTANTS OR EQUILIBRIUM DATA BE SUPPLIED 0053
     1?',/T2,'ENTER ANT OR EQU.')                                       0054
      READ(NCR,2) ENAME                                                 0055
      IF(ENAME.NE.'ANT'.AND.ENAME.NE.'EQU') GO TO 902                   0056
      IF(ENAME.EQ.'EQU') GO TO 502                                      0057
      IF(MCAN.EQ.1) WRITE(NTP,5003)                                     0058
 5003 FORMAT(/T2,'ENTER ANTOINE CONSTANTS.')                            0059
C     ANTOINE CONSTANTS *****************************************************
      NBDC=1                                                            0060
      READ(NCR,1) (A(I),B(I),C(I),I=1,2)                                0061
      IF(MCAN.EQ.1) WRITE(NTP,5004)                                     0062
 5004 FORMAT(/T2,'ENTER BUBBLE PT OF LIGHT COMPONENT.')                 0063
      READ(NCR,1) TBL                                                   0064
      DELXAL=0.02                                                       0065
      XAL=0.0                                                           0066
```

```
      DO 51 J=1,1000                                                    0067
      XAL=XAL+DELXAL                                                    0068
      IF(ABS(XAL-1.0).LT.0.001) GO TO 501                               0069
      CALL BINBPT(A,B,C,NBDC,NDATA,PT,PAL,TALF,TBP,TBL,XALF,XAL)        0070
      YAL=PAL*XAL/PT                                                    0071
      XALF(J)=XAL                                                       0072
      YALF(J)=YAL                                                       0073
      XEQUIL(J+1)=XAL                                                   0074
      YEQUIL(J+1)=YAL                                                   0075
      ALFA(J)=YALF(J)/XALF(J)*(1.-XALF(J))/(1.-YALF(J))                 0076
   51 CONTINUE                                                          0077
  501 NDATA=J-1                                                         0078
      NDAT=J+1                                                          0079
      XEQUIL(1)=0.0                                                     0080
      XEQUIL(NDAT)=1.0                                                  0081
      YEQUIL(1)=0.0                                                     0082
      YEQUIL(NDAT)=1.0                                                  0083
      GO TO 504                                                         0084
  502 CONTINUE                                                          0085
      IF(MCAN.EQ.1) WRITE(NTP,5005)                                     0086
 5005 FORMAT(/T2,'IS THE RELATIVE VOLATILITY A CONSTANT?')              0087
      READ(NCR,2) ANAME                                                 0088
      IF(ANAME.NE.'YES'.AND.ANAME.NE.'NO') GO TO 902                    0089
      IF(ANAME.EQ.'NO') GO TO 503                                       0090
      IF(MCAN.EQ.1) WRITE(NTP,5006)                                     0091
 5006 FORMAT(/T2,'ENTER CONSTANT RELATIVE VOLATILITY VALUE.')           0092
C     CONSTANT RELATIVE VOLATILITY **************************************
      READ(NCR,1) ALFCON                                                0093
      MALF=2                                                            0094
      NDAT=51                                                           0095
      DELX=0.02                                                         0096
      XA=0.0                                                            0097
      DO 52 I=1,NDAT                                                    0098
      XEQUIL(I)=XA                                                      0099
      YEQUIL(I)=XA*ALFCON/(1.+XA*(ALFCON-1.))                           0100
      XA=XA+DELX                                                        0101
   52 CONTINUE                                                          0102
      GO TO 504                                                         0103
```

```
  503 CONTINUE                                                                  0104
      IF(MCAN.EQ.1) WRITE(NTP,5007)                                             0105
 5007 FORMAT(/T2,'ENTER NO. OF SETS OF DATA TO BE SUPPLIED',/T2,'DO NOT         0106
     1INCLUDE X=0.,Y=0. OR X=1.,Y=1.')                                          0107
      READ(NCR,1) NDATA                                                         0108
      IF(MCAN.EQ.1) WRITE(NTP,5008)                                             0109
 5008 FORMAT(/T2,'ADD EQUILIBRIUM-DATA ELEMENT.')                               0110
C     EQUILIBRIUM DATA **********************************************************
      NBDC=2                                                                    0111
      READ(NCR,1) (XALF(I),YALF(I),I=1,NDATA)                                   0112
      DO 53 J=1,NDATA                                                           0113
      ALFA(J)=YALF(J)/XALF(J)*(1.-XALF(J))/(1.-YALF(J))                         0114
   53 CONTINUE                                                                  0115
      XEQUIL(1)=0.0                                                             0116
      YEQUIL(1)=0.0                                                             0117
      NDAT1=NDATA+1                                                             0118
      NDAT=NDATA+2                                                              0119
      DO 54 I=2,NDAT1                                                           0120
      XEQUIL(I)=XALF(I-1)                                                       0121
      YEQUIL(I)=YALF(I-1)                                                       0122
   54 CONTINUE                                                                  0123
      XEQUIL(NDAT)=1.0                                                          0124
      YEQUIL(NDAT)=1.0                                                          0125
  504 CONTINUE                                                                  0126
      IF(NDAT.GT.NPEQ) GO TO 903                                                0127
C     END OF EQUILIBRIUM DATA CALCULATIONS.
C     PROBLEM SPECIFICATIONS.
      IF(MCAN.EQ.1) WRITE(NTP,5009)                                             0128
 5009 FORMAT(/T2,'ENTER NO OF THEORETICAL TRAYS IN COLUMN. (MAX=26)')           0129
      READ(NCR,1) NP                                                            0130
      IF(MCAN.EQ.1) WRITE(NTP,5010)                                             0131
 5010 FORMAT(/T2,'ENTER COMPOSITION AND QUANTITY OF INITIAL CHARGE.')           0132
      READ(NCR,1) XF,FTOT                                                       0133
      IF(XF.GT.1.0.OR.XF.LT.0.0) GO TO 902                                      0134
```

```
      IF(MCAN.EQ.1) WRITE(NTP,5111)                                     0135
 5111 FORMAT(/T2,'WILL (1) BATCH CYCLE TIME,OR (2) VAPOUR BOIL-UP RATE',0136
     1/T2,'BE SPECIFIED. ENTER 1 OR 2,')                                0137
      READ(NCR,1) NTIM                                                  0138
      IF(NTIM.NE.1.AND.NTIM.NE.2) GO TO 902                             0139
      IF(MCAN.EQ.1) WRITE(NTP,5011)                                     0140
 5011 FORMAT(/T2,'ENTER BATCH TIME OR BOIL-UP RATE,')                   0141
      READ(NCR,1) G                                                     0142
      IF(MCAN.EQ.1.AND.NOP1.EQ.1) WRITE(NTP,5012)                       0143
 5012 FORMAT(/T2,'ENTER DISTILLATE COMPOSITION.')                       0144
      IF(NOP1.EQ.1) READ(NCR,1) XD                                      0145
      IF(NOP1.EQ.1) GO TO 505                                           0146
      IF(MCAN.EQ.1) WRITE(NTP,5112)                                     0147
 5112 FORMAT(/T2,'DO YOU WANT TO SPECIFY AN AVERAGE DISTILLATE COMPOSITI 0148
     1ON?')                                                             0149
      READ(NCR,2) DNAME                                                 0150
      IF(DNAME.NE.'YES'.AND.DNAME.NE.'NO') GO TO 902                    0151
      IF(DNAME.EQ.'NO') GO TO 505                                       0152
      IF(MCAN.EQ.1) WRITE(NTP,5113)                                     0153
 5113 FORMAT(T2,'ENTER AVERAGE VALUE.')                                 0154
      READ(NCR,1) XSPEC                                                 0155
      IF(XSPEC.GT.1.0.OR.XSPEC.LT.XF) GO TO 902                         0156
      NSPEC=2                                                           0157
  505 CONTINUE                                                          0158
      IF(NSPEC.EQ.2) GO TO 506                                          0159
      IF(MCAN.EQ.1) WRITE(NTP,5013)                                     0160
 5013 FORMAT(/T2,'ENTER FINAL RESIDUE CONCENTRATION.')                  0161
      READ(NCR,1) XW                                                    0162
      IF(XW.GT.XF.OR.XW.LT.0.0) GO TO 902                               0163
  506 CONTINUE                                                          0164
      IF(MCAN.EQ.1.AND.NOP1.EQ.2) WRITE(NTP,5014)                       0165
 5014 FORMAT(/T2,'ENTER REFLUX RATIO.')                                 0166
      IF(NOP1.EQ.2) READ(NCR,1) R                                       0167
      IF(MCAN.EQ.1) WRITE(NTP,5015)                                     0168
 5015 FORMAT(/T2,'IS A (1) TOTAL OR (2) PARTIAL CONDENSER USED?',       0169
     1/T2,'ENTER 1 OR 2.')                                              0170
      READ(NCR,1) NCOND                                                 0171
```

```
      NPCHEC=NP+NCOND+1                                                 0172
      IF(NPCHEC.GT.NPTC) GO TO 903                                      0173
C     START OF COMPUTATION **********************************************
      IF(NOP1.EQ.2) GO TO 200                                           0174
C
C     VARIABLE REFLUX OPERATION *****************************************
C
C     CALCULATION OF INITIAL REFLUX RATIO ***********************************
C     INITIAL ESTIMATE OF STARTING REFLUX RATIO.
      R=1.1                                                             0175
      DELR=0.1                                                          0176
      MN=1                                                              0177
      NP1=NP+NCOND                                                      0178
C     'GO TO' LOOP 101 SEARCHES FOR CORRECT STARTING REFLUX RATIO.
  101 CONTINUE                                                          0179
      CALL XDSTEP                                                       0180
      IF(ABS(XNTC-XF).LE.0.00001) GO TO 110                             0181
      IF(DELR.LT.0.0000001) GO TO 110                                   0182
      NCALL=2                                                           0183
      CALL SEARCH(XNTC,XF,R,DELR,MN,NCALL)                              0184
      GO TO 101                                                         0185
  110 CONTINUE                                                          0186
C     WRITE INITIAL CONDITIONS TO DATA ELEMENT FOR PLOTTING X-Y DIAGRAM.
      YINCPT=XD/(R+1.)                                                  0187
      WRITE(NG1,8002) NDAT                                              0188
      WRITE(NG1,8001) (XEQUIL(I),YEQUIL(I),I=1,NDAT)                    0189
      WRITE(NG1,8002) NTC                                               0190
      WRITE(NG1,8001) (X(I),Y(I),I=1,NTC)                               0191
      WRITE(NG1,8001) YINCPT,XD                                         0192
C     CALCULATION OF DATA REQUIRED FOR INTEGRATING FUNCTION.
C      FUNCTION: D(VAP)=FTOT*(XD-XF),(D(XW)*(R+1)/(XD-XW)**2))
C     LIMITS OF INTEGRATION: XF,XW ,NO. OF INTERVALS OF SIZE 'H': NINT
      H=(XF-XW)/NINT                                                    0193
      XSTEP=XW-H                                                        0194
      XDATA(NINT+1)=XF                                                  0195
      YDATA(NINT+1)=R                                                   0196
      R=10.                                                             0197
```

```
C     'DO' LOOP 11 COMPUTES NINT RESIDUE COMPOSITION-REFLUX RATIO DATA SETS.
      DO 11 I=1,NINT                                                    0198
      XSTEP=XSTEP+H                                                     0199
      MN=1                                                              0200
      DELR=0.1                                                          0201
C     'GO TO ' LOOP 102 SEARCHES FOR CORRECT R FOR EACH XSTEP VALUE.
  102 CONTINUE                                                          0202
      CALL XDSTEP                                                       0203
      IF(ABS(XNTC-XSTEP).LE.0.00001) GO TO 120                          0204
      IF(DELR.LT.0.000001) GO TO 120                                    0205
      NCALL=2                                                           0206
      CALL SEARCH(XNTC,XSTEP,R,DELR,MN,NCALL)                           0207
      GO TO 102                                                         0208
  120 CONTINUE                                                          0209
      IF(I.GT.1) GO TO 103                                              0210
C     WRITE FINAL CONDITIONS TO DATA ELEMENT FOR PLOTTING X-Y DIAGRAM.
      YINCPT=XD/(R+1.)                                                  0211
      WRITE(NG1,8002) NTC                                               0212
      WRITE(NG1,8001) (X(J),Y(J),J=1,NTC)                               0213
      WRITE(NG1,8001) YINCPT,XD                                         0214
  103 CONTINUE                                                          0215
      XDATA(I)=XSTEP                                                    0216
      YDATA(I)=R                                                        0217
   11 CONTINUE                                                          0218
      NINT1=NINT+1                                                      0219
C     'DO' LOOP 12 CALCULATES VALUES OF THE FUNCTION TO BE INTEGRATED.
      DO 12 J=1,NINT1                                                   0220
      FUNC(J)=(1.+YDATA(J))/((XD-XDATA(J))**2)                          0221
   12 CONTINUE                                                          0222
C     SIMPSONS RULE INTEGRATION OF THE FUNCTION.
      CALL SIMPS(XF,XW,NINT1,FUNC,SUMINT)                               0223
      VAP=FTOT*(XD-XF)*SUMINT                                           0224
      TIME=VAP/G                                                        0225
      W=FTOT*(XD-XF)/(XD-XW)                                            0226
      DTOT=FTOT-W                                                       0227
      GO TO 400                                                         0228
```

MAINLINE BATCH

```
  200 CONTINUE                                                          0229
C
C     CONSTANT REFLUX OPERATION **********************************************
C
C     INITIAL ESTIMATE OF DISTILLATE COMPOSITION XD.
      XD=0.99                                                           0230
      DELX=0.1                                                          0231
      MN=1                                                              0232
      NP1=NP+NCOND                                                      0233
C     'GO TO' LOOP 201 SEARCHES FOR CORRECT STARTING XD.
  201 CONTINUE                                                          0234
      CALL XDSTEP                                                       0235
      IF(ABS(XNTC-XF).LE.0.00001) GO TO 210                             0236
      NCALL=1                                                           0237
      CALL SEARCH(XNTC,XF,XD,DELX,MN,NCALL)                             0238
      IF(XD.GT.1.0) XD=0.999                                            0239
      IF(XD.GT.1.0) DELX=0.000001                                       0240
      GO TO 201                                                         0241
  210 CONTINUE                                                          0242
C     WRITE INITIAL CONDITIONS TO DATA ELEMENT FOR PLOTTING X-Y DIAGRAM.
      YINCPT=XD/(R+1.)                                                  0243
      WRITE(NG1,8002) NDAT                                              0244
      WRITE(NG1,8001) (XEQUIL(I),YEQUIL(I),I=1,NDAT)                    0245
      WRITE(NG1,8002) NTC                                               0246
      WRITE(NG1,8001) (X(I),Y(I),I=1,NTC)                               0247
      WRITE(NG1,8001) YINCPT,XD                                         0248
      XDATA(NINT+1)=XF                                                  0249
      YDATA(NINT+1)=XD                                                  0250
      DELXW=0.01                                                        0251
      MM=1                                                              0252
      IF(NSPEC.EQ.2) XW=0.1                                             0253
```

```
C     'GO TO' LOOP 204 CONSTRAINS SYSTEM TO SPECIFIED AVERAGE XD VALUE.
  204 CONTINUE                                                          0254
C     CALCULATION OF DATA REQUIRED FOR INTEGRATING FUNCTION.
C      FUNCTION: D(ALOG(FTOT/W)=D(XW)/(XD-XW)
C     LIMITS OF INTEGRATION: XF 'XW 'NO. OF INTERVALS OF SIZE 'H': NINT
      H=(XF-XW)/NINT                                                    0255
      XSTEP=XW-H                                                        0256
      XD=0.99                                                           0257
C     'DO' LOOP 21 COMPUTES NINT RESIDUE-DISTILLATE COMPOSITION DATA SETS.
      DO 21 I=1,NINT                                                    0258
      XSTEP=XSTEP+H                                                     0259
      MN=1                                                              0260
      DELX=0.1                                                          0261
C     'GO TO ' LOOP 202 SEARCHES FOR CORRECT XD FOR EACH XSTEP VALUE.
  202 CONTINUE                                                          0262
      CALL XDSTEP                                                       0263
      IF(ABS(XNTC-XSTEP).LE.0.00001) GO TO 220                          0264
      NCALL=1                                                           0265
      CALL SEARCH(XNTC,XSTEP,XD,DELX,MN,NCALL)                          0266
      IF(XD.GT.1.0) XD=0.999                                            0267
      IF(XD.GT.1.0) DELX=0.000001                                       0268
      GO TO 202                                                         0269
  220 CONTINUE                                                          0270
      IF(I.GT.1) GO TO 203                                              0271
C     SAVE FINAL CONDITIONS FOR EACH PASS THROUGH LOOP 204.
C     XEQUIL,YEQUIL ARE USED HERE AS STORAGE SPACE.
      YINCPT=XD/(R+1.)                                                  0272
      DO 20 J=1,NTC                                                     0273
      XEQUIL(J)=X(J)                                                    0274
      YEQUIL(J)=Y(J)                                                    0275
   20 CONTINUE                                                          0276
      XDHOLD=XD                                                         0277
      NTCH=NTC                                                          0278
  203 CONTINUE                                                          0279
      XDATA(I)=XSTEP                                                    0280
      YDATA(I)=XD                                                       0281
   21 CONTINUE                                                          0282
```

```
      NINT1=NINT+1                                                          0283
C     'DO' LOOP 22 CALCULATES VALUES OF THE FUNCTION TO BE INTEGRATED.
      DO 22 J=1,NINT1                                                       0284
      FUNC(J)=1./(YDATA(J)-XDATA(J))                                        0285
   22 CONTINUE                                                              0286
C     SIMPSONS RULE INTEGRATION OF THE FUNCTION.
      CALL SIMPS(XF,XW,NINT1,FUNC,SUMINT)                                   0287
      W=FTOT/(EXP(SUMINT))                                                  0288
      DTOT=FTOT-W                                                           0289
      VAP=DTOT*(R+1.)                                                       0290
      TIME=VAP/G                                                            0291
      XDAVE=(FTOT*XF-W*XW)/DTOT                                             0292
      IF(NSPEC.EQ.1) GO TO 230                                              0293
      IF(ABS(XDAVE-XSPEC).LT.0.0001) GO TO 230                              0294
      NCALL=1                                                               0295
      CALL SEARCH(XDAVE,XSPEC,XW,DELXW,MM,NCALL)                            0296
      IF(XW.LT.0.001) GO TO 901                                             0297
      GO TO 204                                                             0298
  230 CONTINUE                                                              0299
      WRITE(NG1,8002) NTCH                                                  0300
      WRITE(NG1,8001) (XEQUIL(J),YEQUIL(J),J=1,NTCH)                        0301
      WRITE(NG1,8001) YINCPT,XDHOLD                                         0302
      WRITE(NG1,8001) XDAVE,XD                                              0303
  400 CONTINUE                                                              0304
C     PRINT RESULTS ***************************************************
      WRITE(NTP,4100)                                                       0305
 4100 FORMAT(/T2,'BATCH DISTILLATION OF A BINARY SOLUTION,')                0306
      IF(NOP1.EQ.1) WRITE(NTP,4101)                                         0307
      IF(NOP1.EQ.2) WRITE(NTP,4102)                                         0308
 4101 FORMAT(T2,'VARIABLE REFLUX OPERATION,')                               0309
 4102 FORMAT(T2,'CONSTANT REFLUX OPERATION,')                               0310
```

```
C      COMPONENT TITLES.
       WRITE(NTP,4103) ((TITLES(I,J),J=1,5),I=1,2)                      0311
 4103 FORMAT(/T2,'FEED COMPONENTS: (1) ',5A6,/T19,'(2) ',5A6)           0312
C      VARIABLE REFLUX OPERATION: INITIAL AND FINAL REFLUX RATIOS.
       IF(NOP1.EQ.1) WRITE(NTP,4104) YDATA(NINT1),YDATA(1)              0313
 4104 FORMAT(/T2,'INITIAL REFLUX RATIO:',F7.3,/T2,'FINAL REFLUX RATIO:',0314
     1F9.3)                                                             0315
C      CONSTANT REFLUX OPERATION: REFLUX RATIO.
       IF(NOP1.EQ.2) WRITE(NTP,4105) R                                  0316
 4105 FORMAT(/T2,'REFLUX RATIO:',F7.3)                                  0317
C      VAPOUR BOIL-UP RATE AND BATCH CYCLE TIME.
       IF(NTIM.EQ.1) WRITE(NTP,4106) TIME,G                             0318
       IF(NTIM.EQ.2) WRITE(NTP,4106) G,TIME                             0319
 4106 FORMAT(/T2,'VAPOUR BOIL-UP RATE:',F6.2,2X,'KGMOLE/HR',            0320
     1/T2,'BATCH CYCLE TIME:',F10.3,1X,'HRS')                           0321
       WRITE(NTP,4108) VAP                                              0322
 4108 FORMAT(/T2,'TOTAL VAPOUR BOILED UP DURING CYCLE:',F8.3,' KGMOLE') 0323
C      RESIDUE DATA.
       WRITE(NTP,4109) W,XW                                             0324
 4109 FORMAT(/T2,'RESIDUE:',9X,F8.3,' KGMOLE',/T2,'RESIDUE COMPOSITION:'0325
     1,F6.4)                                                            0326
C      DISTILLATE PRODUCT DATA.
       WRITE(NTP,4110) DTOT                                             0327
 4110 FORMAT(/T2,'TOTAL DISTILLATE WITHDRAWN:',F8.3,' KGMOLE')          0328
       IF(NOP1.EQ.1) WRITE(NTP,4111) XD                                 0329
 4111 FORMAT(/T2,'DISTILLATE COMPOSITION:',F6.4,' MOLE FRAC')           0330
       IF(NOP1.EQ.2) WRITE(NTP,4112) YDATA(NINT1),YDATA(1),XDAVE        0331
 4112 FORMAT(/T2,'INITIAL DISTILLATE COMPOSITION:',F6.4,' MOLE FRAC',/T2 0332
     1,'FINAL DISTILLATE COMPOSITION:',F8.4,/T2,                        0333
     2'AVERAGE DISTILLATE COMPOSITION:',F6.4)                           0334
       IF(NOP1.EQ.2) GO TO 401                                          0335
```

```
C       VARIABLE REFLUX OPERATION: RESIDUE COMPOSITION VS REFLUX RATIO,
        WRITE(NTP,4113)                                                  0336
 4113   FORMAT(/T2,'RESIDUE COMPOSITION VS REFLUX RATIO:',/T6,'XW',9X,'R')  0337
        WRITE(NTP,4114) (XDATA(I),YDATA(I),I=NINT1,1,-10)                0338
 4114   FORMAT(T3,F6.4,3X,F8.4)                                          0339
        GO TO 999                                                        0340
  401   CONTINUE                                                         0341
C       CONSTANT REFLUX OPERATION: RESIDUE COMP VS DISTILLATE COMP,
        WRITE(NTP,4115)                                                  0342
 4115   FORMAT(/T2,'RESIDUE COMPOSITION VS DISTILLATE COMPOSITION:',/T6,  0343
     1'XW',9X,'XD')                                                      0344
        WRITE(NTP,4114) (XDATA(I),YDATA(I),I=NINT1,1,-10)                0345
        GO TO 999                                                        0346
C       ERROR MESSAGES ***************************************************
  901   CONTINUE                                                         0347
        WRITE(NTP,9001) XW                                               0348
 9001   FORMAT(/T2,'RESIDUE COMPOSITION:',F8.4,/T2,'DISTILLATE COMPOSITION  0349
     1 SET TOO LOW.')                                                    0350
        GO TO 999                                                        0351
  902   CONTINUE                                                         0352
        WRITE(NTP,9002)                                                  0353
 9002   FORMAT(/T2,'SPECIFICATION ERROR...RESTART PROGRAM.')             0354
        GO TO 999                                                        0355
  903   CONTINUE                                                         0356
        WRITE(NTP,9003)                                                  0357
 9003   FORMAT(/T2,'DIMENSION SPECIFICATIONS EXCEEDED.')                 0358
  999   CONTINUE                                                         0359
        WRITE(NTP,9991)                                                  0360
 9991   FORMAT(/T2,'END OF PROGRAM.')                                    0361
        END                                                              0362
```

MAINLINE BATCH

```
      SUBROUTINE XDSTEP                                                  0001
C     ROUTINE STEPS OFF MCCABE-THIELE STAGES IN THE DESIGN OF BINARY
C     BATCH DISTILLATION SYSTEMS.
      PARAMETER NPEQ=55                                                  0002
      PARAMETER NPTC=55                                                  0003
      COMMON ALFA(NPEQ),ALFCON,R                                         0004
      COMMON MALF,NDATA,NP1,NTC                                          0005
      COMMON X(NPTC),XALF(NPEQ),XD,XNTC,Y(NPTC),YALF(NPEQ)               0006
      X(1)=XD                                                            0007
      Y(1)=XD                                                            0008
      MSTOP=NP1*2                                                        0009
      DO 11 I=2,100,2                                                    0010
      Y(I)=Y(I-1)                                                        0011
      YA=Y(I)                                                            0012
      CALL XYALFA(ALFA,ALFCON,ALF,MALF,NDATA,YALF,YA)                    0013
      X(I)=Y(I)/(ALF+Y(I)*(1.-ALF))                                      0014
      IF(I.EQ.MSTOP) GO TO 110                                           0015
      Y(I+1)=X(I)*R/(R+1.)+XD/(R+1.)                                     0016
      X(I+1)=X(I)                                                        0017
   11 CONTINUE                                                           0018
  110 CONTINUE                                                           0019
      Y(I+1)=0.0                                                         0020
      X(I+1)=X(I)                                                        0021
      XNTC=X(I)                                                          0022
      NTC=I+1                                                            0023
      RETURN                                                             0024
      END                                                                0025
```

```
      SUBROUTINE BINBPT(A,B,C,N,NDATA,PT,P1,TALF,TBP,T1,XALF,X1)          0001
C     BUBBLE PT. ROUTINE FOR BINARY SYSTEMS.
      DIMENSION A(2),B(2),C(2),TALF(1),TBP(2),XALF(1),XBPT(2)            0002
      GO TO (100,200),N                                                  0003
C     BUBBLE PT BY NEWTON'S METHOD,USING ANTOINE CONSTANTS.
  100 CONTINUE                                                           0004
      T=T1                                                               0005
      XBPT(1)=X1                                                         0006
      XBPT(2)=1.-X1                                                      0007
      DO 10 I=1,1000                                                     0008
      SUM1=PT                                                            0009
      SUM2=0.0                                                           0010
      DO 11 K=1,2                                                        0011
      A1=A(K)-B(K)/(C(K)+T)                                              0012
      SUM1=SUM1-XBPT(K)*10.**A1                                          0013
      B1=2.303*B(K)/((C(K)+T)**2.)                                       0014
      SUM2=SUM2+XBPT(K)*B1*10.**A1                                       0015
   11 CONTINUE                                                           0016
      DIFF=SUM1/SUM2                                                     0017
      IF(ABS(DIFF).LT.0.00001) GO TO 101                                 0018
      T=T+DIFF                                                           0019
   10 CONTINUE                                                           0020
  101 CONTINUE                                                           0021
      FACT=A(1)-B(1)/(C(1)+T)                                            0022
      T1=T                                                               0023
      P1=10.**FACT                                                       0024
      GO TO 999                                                          0025
```

```
C     BUBBLE PT SEARCH FROM SUPPLIED VAPOUR-LIQUID EQUILIBRIUM DATA.
  200 CONTINUE                                                          0026
      IF(X1.GT.XALF(1)) GO TO 201                                       0027
      XFACT=X1/XALF(1)                                                  0028
      T1=TBP(2)+(TALF(1)-TBP(2))*XFACT                                  0029
      GO TO 999                                                         0030
  201 CONTINUE                                                          0031
      IF(X1.GT.XALF(NDATA)) GO TO 220                                   0032
      DO 20 I=1,NDATA                                                   0033
      DIFF=X1-XALF(I)                                                   0034
      IF(DIFF.LE.0.0) GO TO 210                                         0035
   20 CONTINUE                                                          0036
      GO TO 220                                                         0037
  210 XFACT=(XALF(I)-X1)/(XALF(I)-XALF(I-1))                            0038
      T1=TALF(I)+(TALF(I-1)-TALF(I))*XFACT                              0039
      GO TO 999                                                         0040
  220 CONTINUE                                                          0041
      XFACT=(1.-X1)/(1.-XALF(NDATA))                                    0042
      T1=TBP(1)+(TALF(NDATA)-TBP(1))*XFACT                              0043
  999 CONTINUE                                                          0044
      RETURN                                                            0045
      END                                                               0046
```

SUBROUTINE BINBPT

```
      SUBROUTINE BINDPT(A,B,C,N,NDATA,PT,P1,TALF,TBP,T1,YALF,Y1)          0001
C     DEW POINT ROUTINE FOR BINARY SYSTEMS.
      DIMENSION A(2),B(2),C(2),TALF(1),TBP(2),YALF(1),YDPT(2)            0002
      GO TO (100,200),N                                                  0003
C     DEW PT BY NEWTON'S METHOD,USING ANTOINE CONSTANTS.
  100 CONTINUE                                                           0004
      T=T1                                                               0005
      YDPT(1)=Y1                                                         0006
      YDPT(2)=1.-Y1                                                      0007
      DO 10 I=1,1000                                                     0008
      SUM1=1.                                                            0009
      SUM2=0.0                                                           0010
      DO 11 K=1,2                                                        0011
      A1=A(K)-B(K)/(C(K)+T)                                              0012
      A2=10.**A1                                                         0013
      SUM1=SUM1-YDPT(K)*PT/A2                                            0014
      B1=2.303*B(K)/((C(K)+T)**2.)                                       0015
      SUM2=SUM2+YDPT(K)*B1*PT/A2                                         0016
   11 CONTINUE                                                           0017
      DIFF=SUM1/SUM2                                                     0018
      IF(ABS(DIFF).LT.0.00001) GO TO 101                                 0019
      T=T-DIFF                                                           0020
   10 CONTINUE                                                           0021
  101 CONTINUE                                                           0022
      FACT=A(1)-B(1)/(C(1)+T)                                            0023
      T1=T                                                               0024
      P1=10.**FACT                                                       0025
      GO TO 999                                                          0026
```

```
C     DEW PT SEARCH FROM SUPPLIED VAPOUR-LIQUID EQUILIBRIUM DATA.
  200 CONTINUE                                                          0027
      IF(Y1.GT.YALF(1)) GO TO 201                                       0028
      YFACT=Y1/YALF(1)                                                  0029
      T1=TBP(2)+(TALF(1)-TBP(2))*YFACT                                  0030
      GO TO 999                                                         0031
  201 CONTINUE                                                          0032
      IF(Y1.GT.YALF(NDATA)) GO TO 220                                   0033
      DO 20 I=1,NDATA                                                   0034
      DIFF=Y1-YALF(I)                                                   0035
      IF(DIFF.LE.0.0) GO TO 210                                         0036
   20 CONTINUE                                                          0037
      GO TO 220                                                         0038
  210 YFACT=(YALF(I)-Y1)/(YALF(I)-YALF(I-1))                            0039
      T1=TALF(I)+(TALF(I-1)-TALF(I))*YFACT                              0040
      GO TO 999                                                         0041
  220 CONTINUE                                                          0042
      YFACT=(1.-Y1)/(1.-YALF(NDATA))                                    0043
      T1=TBP(1)+(TALF(NDATA)-TBP(1))*YFACT                              0044
  999 CONTINUE                                                          0045
      RETURN                                                            0046
      END                                                               0047
```

SUBROUTINE BINDPT

```
      SUBROUTINE XYALFA(ALFA,ALFCON,ALPHA,MALF,NDATA,XYALF,XY1)
C     ROUTINE SEARCHES FOR THE RELATIVE VOLATILITY CORRESPONDING
C     TO THE SUPPLIED LIQUID OR VAPOUR COMPOSITION VALUE.
      DIMENSION ALFA(1),XYALF(1)
      IF(MALF.GT.1) GO TO 200
      IF(XY1.GT.XYALF(1)) GO TO 100
      XYFACT=(XYALF(1)-XY1)/XYALF(1)
      ALPHA=ALFA(1)-(ALFA(2)-ALFA(1))*XYFACT
      GO TO 999
  100 CONTINUE
      IF(XY1.GT.XYALF(NDATA)) GO TO 120
      DO 10 I=1,NDATA
      DIFF=XY1-XYALF(I)
      IF(DIFF.LE.0.0) GO TO 110
   10 CONTINUE
      GO TO 120
  110 XYFACT=(XYALF(I)-XY1)/(XYALF(I)-XYALF(I-1))
      ALPHA=ALFA(I)+(ALFA(I-1)-ALFA(I))*XYFACT
      GO TO 999
  120 CONTINUE
      XYFACT=(1.0-XY1)/(1.0-XYALF(NDATA))
      ALPHA=1.0+(ALFA(NDATA)-1.0)*XYFACT
      GO TO 999
  200 ALPHA=ALFCON
      GO TO 999
  999 CONTINUE
      RETURN
      END
```

SUBROUTINE XYALFA

```
      SUBROUTINE SIMPS(ALIM,BLIM,N,FUNC,SUM)                            0001
C     ROUTINE INTEGRATES A FUNCTION OVER M INTERVALS BETWEEN LIMITS
C     ALIM AND BLIM BY SIMPSON'S RULE. VALUES OF THE FUNCTION ARE
C     STORED IN ARRAY 'FUNC'. INTEGRAL RETURNED AS 'SUM'.
      DIMENSION FUNC(1)                                                 0002
      M=N-1                                                             0003
      EVENS=0.0                                                         0004
      ODDS=0.0                                                          0005
      H=(ALIM-BLIM)/M                                                   0006
      DO 10 J=2,M                                                       0007
      IF(MOD(J,2)) 102,101,102                                          0008
  101 EVENS=EVENS+FUNC(J)                                               0009
      GO TO 10                                                          0010
  102 ODDS=ODDS+FUNC(J)                                                 0011
   10 CONTINUE                                                          0012
      SUM=H*(FUNC(1)+FUNC(N)+4.*EVENS+2.*ODDS)/3.                       0013
      RETURN                                                            0014
      END                                                               0015
```

```
      SUBROUTINE SEARCH(FA1,FA2,FACTOR,DEL,MM,N)                        0001
C     ROUTINE RETURNS A NEW ESTIMATE OF 'C' TO BE USED BY THE MAINLINE
C     TO CONVERGE THE EQUALITY A=B. 'DEL' IS INCREMENT IN C THAT DECREASES
C     AS CONVERGENCE IS APPROACHED.
C     'MM' CAUSES DEL TO BE DECREASED AS A SWITCHES FROM >B TO <B.
C     'N' DETERMINES THE EFFECT OF C ON CONVERGENCE.
      IF(FA1.LT.FA2) GO TO 120                                          0002
      IF(MM.EQ.1) GO TO 111                                             0003
      DEL=DEL/2.                                                        0004
      MM=1                                                              0005
  111 IF(N.EQ.1) FACTOR=FACTOR-DEL                                      0006
      IF(N.EQ.2) FACTOR=FACTOR+DEL                                      0007
      GO TO 999                                                         0008
  120 IF(MM.EQ.2) GO TO 121                                             0009
      DEL=DEL/2.                                                        0010
      MM=2                                                              0011
  121 IF(N.EQ.1) FACTOR=FACTOR+DEL                                      0012
      IF(N.EQ.2) FACTOR=FACTOR-DEL                                      0013
  999 CONTINUE                                                          0014
      RETURN                                                            0015
      END                                                               0016
```

```
C       COMMON BLOCK 'BINCOM' FOR BINARY DISTILLATION PROGRAMS.
        PARAMETER NPEQ=55                                                       0001
        PARAMETER NPPS=150                                                      0002
        PARAMETER NPTC=55                                                       0003
        PARAMETER NPEN=25                                                       0004
        DIMENSION ALFA(NPEQ),HLIQ(NPEN),HVAP(NPEN),LIQR(NPPS),LIQS(NPPS)        0005
        DIMENSION LRWT(NPPS),LSWT(NPPS),TALF(NPEQ)                              0006
        DIMENSION TRL(NPPS),TRV(NPPS),TSL(NPPS),TSV(NPPS)                       0007
        DIMENSION VAPR(NPPS),VAPS(NPPS),VRWT(NPPS),VSWT(NPPS)                   0008
        DIMENSION X(NPTC),XALF(NPEQ),XEQUIL(NPEQ),XH(NPEN)                      0009
        DIMENSION XROP(NPPS),XSOP(NPPS),YROP(NPPS),YSOP(NPPS)                   0010
        DIMENSION Y(NPTC),YALF(NPEQ),YEQUIL(NPEQ),YH(NPEN)                      0011
        COMMON A(2),ACP(2),AL(11),ALFA,ALFCON,AV(11)                            0012
        COMMON B(2),BCP(2),BTOT,C(2),CCP(2),CFCT(12),CPCONV,CPMW(2)             0013
        COMMON DCP(2),DELB,DELD,DL,DTOT,DV                                      0014
        COMMON EFF,FLOW(10),FTOT,FTOTS(5)                                       0015
        COMMON HB,HD,HEVABS,HF,HLIQ,HSCONV,HSMW(2),HSOLN(30)                    0016
        COMMON HVAP,LH(2)                                                       0017
        COMMON LHCONV,LHMW(2),LIQR,LIQS,LRWT,LSWT                               0018
        COMMON MALF,MW(2),NBDC,NCOND,NDATA,NENTH,NF,NHS,NINT                    0019
        COMMON NN,NOP,NOS,NPPSC,NPTCC,NS,NTRAY(12)                              0020
        COMMON NCR,NG1,NTP                                                      0021
        COMMON PT,Q,QB,QC,QF(5),QS(5),QFS(10),QX(12),QY(12),R,RM                0022
        COMMON SLPE(12),STOTS(5),TALF,TBASE,TBP(2),TC(2)                        0023
        COMMON TCONV,TF,TG,TITLES(2,5),TRAYNO,TRL,TRV                           0024
        COMMON TSL,TSV,VAPR,VAPS,VRWT,VSWT                                      0025
        COMMON X,XALF,XB,XC,XCR,XD,XEQUIL,XH,XHS(30)                            0026
        COMMON XROP,XSOP,YROP,YSOP                                              0027
        COMMON Y,YALF,YB,YC,YCR,YD,YEQUIL,YH                                    0028
        COMMON Z,ZF(5),ZF1(5),ZS(5),ZS1(5),ZFS(10)                              0029
```

```
C     MAINLINE PROGRAM FOR SOLUTION OF BINARY CONTINUOUS-DISTILLATION
C     PROBLEMS BY THE MCCABE-THIELE OR PONCHON-SAVARIT GRAPHICAL METHOD.
C     THE PROGRAM SETS UP THE EQUILIBRIUM CURVE AND OPERATING LINES
C     FOR THE GIVEN SYSTEM,AND STEPS OFF THE NUMBER OF STAGES REQUIRED
C     FOR THE SPECIFIED SEPARATION.
C     AN OUT-PUT FILE OF DATA IS GENERATED THAT CAN BE USED TO PRODUCE
C     A MCCABE-THIELE OR PONCHON-SAVARIT DIAGRAM FOR THE SYSTEM,ON THE
C     'CALCOMP' PLOTTER.
C     ***********************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     ***********************************
      REAL LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                        0001
    1 FORMAT( )                                                         0002
    2 FORMAT(A3)                                                        0003
    3 FORMAT(5A6)                                                       0004
      DATA MALF'MCAN,NCODE,NINT/1,1,2,50/                               0005
      DATA NCR,NGI,NTP/8,11,5/                                          0006
      NPPSC=NPPS-2                                                      0007
      NPTCC=NPTC                                                        0008
C     START OF PROGRAM ***************************************************
      WRITE(NTP,5000)                                                   0009
 5000 FORMAT(/T2,'ANSWER ALL QUESTIONS YES OR NO.',/T2,'ALL INPUT IN FRE 0010
     1E FORMAT.',//T2,'DO YOU KNOW HOW TO USE THIS PROGRAM?')           0011
      READ(NCR,2) ANAME                                                 0012
      IF(ANAME.NE.'YES'.AND.ANAME.NE.'NO') GO TO 901                    0013
      IF(ANAME.EQ.'YES') GO TO 500                                      0014
      CALL TEACH(NCODE)                                                 0015
      GO TO 999                                                         0016
  500 CONTINUE                                                          0017
      WRITE(NTP,5010)                                                   0018
 5010 FORMAT(/T2,'DESIGN TECHNIQUES: (1) MCCABE-THIELE',/T21,           0019
     1'(2) PONCHON-SAVARIT',/T2,'ENTER 1 OR 2 FOR REQUIRED PROCEDURE,') 0020
      READ(NCR,1) NOP1                                                  0021
      IF(NOP1.EQ.1) NOP=1                                               0022
      IF(NOP1.EQ.2) NOP=3                                               0023
```

```
      WRITE(NTP,5011)                                                    0024
 5011 FORMAT(/T2,'DO YOU WANT TO ADD A DATA ELEMENT?')                   0025
      READ(NCR,2) CNAME                                                  0026
      IF(CNAME.NE.'YES'.AND.CNAME.NE.'NO') GO TO 901                     0027
      IF(CNAME.EQ.'NO') GO TO 501                                        0028
      MCAN=2                                                             0029
      WRITE(NTP,5012)                                                    0030
 5012 FORMAT(/T2,'ADD YOUR DATA ELEMENT.')                               0031
  501 CONTINUE                                                           0032
C     'READBD' READS DATA ELEMENT.
      CALL READBD(MCAN,NDAT)                                             0033
      IF(MALF.GT.999) GO TO 999                                          0034
C     START OF COMPUTATION **********************************************
      IF(NOP.GT.1) GO TO 301                                             0035
C
C     MCCABE-THIELE DESIGN TECHNIQUE.
C
      IF(NF.LT.2.AND.NS.EQ.0) GO TO 101                                  0036
C     COMPLEX COLUMNS:  IE. MORE THAN ONE FEED,AND/OR SIDE STREAMS.
C     'MULTFS' COMPUTES OPERATING CONDITIONS FOR COMPLEX COLUMNS.
      CALL MULTFS(NN1)                                                   0037
      IF(MALF.GT.999) GO TO 999                                          0038
      GO TO 302                                                          0039
  101 CONTINUE                                                           0040
C     SIMPLE COLUMNS:  IE. ONLY ONE FEED,NO SIDE STREAMS.
C     'SIMPLE' COMPUTES OPERATING CONDITIONS FOR SIMPLE COLUMNS.
      CALL SIMPLE                                                        0041
      IF(MALF.GT.999) GO TO 999                                          0042
      NN1=3                                                              0043
      GO TO 302                                                          0044
```

```
  301 CONTINUE                                                          0045
C
C     PONCHON-SAVARIT DESIGN TECHNIQUE.
C
C     'PONSAV' CONSTRUCTS THE PONCHON-SAVARIT OPERATING DIAGRAM.
      CALL PONSAV(NPTS,MPTS,NFIN,MFIN)                                  0046
      IF(MALF.GT.999) GO TO 999                                         0047
      NN1=3                                                             0048
  302 CONTINUE                                                          0049
C     CONSTRUCTION OF STEPPED STAGES ON X-Y DIAGRAM **********************
C     'CHEC' ENSURES THAT NO PINCH POINTS WILL BE ENCOUNTERED.
      CALL CHEC(ALFA,ALFCON,FLOW,MALF,NDATA,NN,NTP,XALF,QX,QY,ZFS)      0050
      IF(MALF.GT.999) GO TO 999                                         0051
C     'STEPD' STEPS OFF STAGES ON X-Y DIAGRAM AND COMPUTES:
      CALL STEPD(NII,NPTS,MPTS)                                         0052
C     PRINT RESULTS ******************************************************
C     'BINRES' PRINTS RESULTS OF RUN.
      CALL BINRES(NDAT,NII,NN1,NPTS,MPTS)                               0053
C
C     WRITE DATA TO FILE FOR PLOTTING X-Y AND ENTHALPY-CONCENTRATION
C     DIAGRAMS.
C
 8001 FORMAT(2(2X,F6.4))                                                0054
 8002 FORMAT(I5)                                                        0055
 8003 FORMAT(F7.4,F10.2)                                                0056
 8004 FORMAT(2E12.6)                                                    0057
 8005 FORMAT(F6.4,E12.6)                                                0058
      IF(NOP.EQ.1) GO TO 800                                            0059
C     ENTHALPY-CONCENTRATION DIAGRAM DATA.
      WRITE(NG1,8004) DELD,DELB                                         0060
      WRITE(NG1,8005) XD,HD                                             0061
      WRITE(NG1,8005) XB,HB                                             0062
      WRITE(NG1,8005) Z,HF                                              0063
      WRITE(NG1,8002) NENTH                                             0064
      WRITE(NG1,8005) (XH(K),HLIQ(K),K=1,NENTH)                         0065
      WRITE(NG1,8005) (YH(K),HVAP(K),K=1,NENTH)                         0066
```

```
  800 CONTINUE                                                          0067
C     EQUILIBRIUM DATA.
      WRITE(NG1,8002) NDAT                                              0068
      WRITE(NG1,8001) (XEQUIL(K),YEQUIL(K),K=1,NDAT)                    0069
      IF(NOP.EQ.3) GO TO 801                                            0070
C     POINTS OF INTERSECTION OF MCCABE-THIELE OPERATING LINES.
      WRITE(NG1,8002) NN1                                               0071
      WRITE(NG1,8001) (QX(K),QY(K),K=1,NN1)                             0072
      GO TO 802                                                         0073
  801 CONTINUE                                                          0074
C     OPERATING LINES FOR PONCHON-SAVARIT DESIGNS.
      WRITE(NG1,8002) NPTS                                              0075
      WRITE(NG1,8001) (XROP(N),YROP(N),N=1,NPTS)                        0076
      WRITE(NG1,8002) MPTS                                              0077
      WRITE(NG1,8001) (XSOP(M),YSOP(M),M=1,MPTS)                        0078
  802 CONTINUE                                                          0079
C     POINTS DEFINING EQUILIBRIUM STAGE STEPS ON X,Y DIAGRAM.
      WRITE(NG1,8002) NII                                               0080
      WRITE(NG1,8001) (X(K),Y(K),K=1,NII)                               0081
      WRITE(NG1,8002) NN                                                0082
C     FEED AND SIDE STREAM COMPOSITIONS AND FLOWRATES (COMPLEX COLUMN).
      IF(NN.GT.2) WRITE(NG1,8003) (ZFS(K),FLOW(K),K=2,NN)               0083
C     FEED STREAM COMPOSITION AND FLOWRATE (SIMPLE COLUMN).
      IF(NN.LE.2) WRITE(NG1,8003) Z,FTOT                                0084
C     POINT OF INTERSECTION OF PONCHON-SAVARIT OPERATING LINES.
      IF(NOP.EQ.3) WRITE(NG1,8001) XCR,YCR                              0085
      GO TO 999                                                         0086
C     ERROR MESSAGES ************************************************
  901 CONTINUE                                                          0087
      WRITE(NTP,9001)                                                   0088
 9001 FORMAT(/T2,'ILLEGAL SPECIFICATION. ERROR TERMINATION.')           0089
  999 CONTINUE                                                          0090
      WRITE(NTP,9999)                                                   0091
 9999 FORMAT(/T2,'END OF PROGRAM.')                                     0092
      STOP                                                              0093
      END                                                               0094
```

MAINLINE BINARY

```
      SUBROUTINE READBD(MCAN,NDAT)                                          0001
C     ROUTINE FOR READING DATA FROM EITHER FILE OR TERMINAL,FOR
C     BINARY CONTINUOUS-DISTILLATION PROBLEMS.
C     ************************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     ************************************
      REAL LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                            0002
    1 FORMAT( )                                                             0003
    2 FORMAT(A3)                                                            0004
    3 FORMAT(5A6)                                                           0005
      DATA ATM,FACTP1,FACTP2/14.696,51.714752,7.50061683/                   0006
      IF(MCAN.EQ.1) WRITE(NTP,5000)                                         0007
 5000 FORMAT(/T2,'ENTER NAMES OF COMPONENTS...1 PER LINE,MAX 30 CHARACTE    0008
     1RS EACH,')                                                            0009
      READ(NCR,3) ((TITLES(I,J),J=1,5),I=1,2)                               0010
      IF(MCAN.EQ.1) WRITE(NTP,5001)                                         0011
 5001 FORMAT(T4,'ACCEPTED PRESSURE UNITS: 1=MMHG,2=PSIG,3=PSIA,4=KPA,5=A    0012
     1TM.',/T2,'ENTER CONDENSER PRESSURE AND UNIT CODE, (EG, 760.0,1)')     0013
      READ(NCR,1) PCONDA,NUNITP                                             0014
      IF(NUNITP.EQ.1) PT=PCONDA                                             0015
      IF(NUNITP.EQ.2) PT=(PCONDA+ATM)*FACTP1                                0016
      IF(NUNITP.EQ.3) PT=PCONDA*FACTP1                                      0017
      IF(NUNITP.EQ.4) PT=PCONDA*FACTP2                                      0018
      IF(NUNITP.EQ.5) PT=PCONDA*ATM*FACTP1                                  0019
      IF(PT.GT.7600.) GO TO 901                                             0020
      IF(MCAN.EQ.1) WRITE(NTP,5002)                                         0021
 5002 FORMAT(/T2,'WILL ANTOINE CONSTANTS OR EQUILIBRIUM DATA BE SUPPLIED    0022
     1?',/T2,'ENTER ANT OR EQU,')                                           0023
      READ(NCR,2) ENAME                                                     0024
      IF(ENAME.NE.'ANT'.AND.ENAME.NE.'EQU') GO TO 901                       0025
      IF(ENAME.EQ.'EQU') GO TO 502                                          0026
      IF(MCAN.EQ.1) WRITE(NTP,5003)                                         0027
 5003 FORMAT(/T2,'ENTER ANTOINE CONSTANTS.')                                0028
C     ANTOINE CONSTANTS ***************************************************
      NBDC=1                                                                0029
      READ(NCR,1) (A(I),B(I),C(I),I=1,2)                                    0030
```

```
      IF(MCAN.EQ.1) WRITE(NTP,5004)                                          0031
 5004 FORMAT(/T2,'ENTER BUBBLE POINTS OF PURE COMPONENTS.')                  0032
      READ(NCR,1) (TBP(I),I=1,2)                                             0033
      DELXAL=0.02                                                            0034
      XAL=0.0                                                                0035
      TBL=(TBP(1)+TBP(2))/2.                                                 0036
      DO 51 J=1,1000                                                         0037
      XAL=XAL+DELXAL                                                         0038
      IF(ABS(XAL-1.0).LT.0.001) GO TO 501                                    0039
      CALL BINBPT(A,B,C,NBDC,NDATA,PT,PAL,TALF,TBP,TBL,XALF,XAL)             0040
      YAL=PAL*XAL/PT                                                         0041
      XALF(J)=XAL                                                            0042
      YALF(J)=YAL                                                            0043
      TALF(J)=TBL                                                            0044
      XEQUIL(J+1)=XAL                                                        0045
      YEQUIL(J+1)=YAL                                                        0046
      ALFA(J)=YALF(J)/XALF(J)*(1.-XALF(J))/(1.-YALF(J))                      0047
   51 CONTINUE                                                               0048
  501 NDATA=J-1                                                              0049
      NDAT=J+1                                                               0050
      XEQUIL(1)=0.0                                                          0051
      XEQUIL(NDAT)=1.0                                                       0052
      YEQUIL(1)=0.0                                                          0053
      YEQUIL(NDAT)=1.0                                                       0054
      GO TO 504                                                              0055
  502 CONTINUE                                                               0056
      IF(NOP.GT.1) GO TO 503                                                 0057
      IF(MCAN.EQ.1) WRITE(NTP,5005)                                          0058
 5005 FORMAT(/T2,'IS THE RELATIVE VOLATILITY A CONSTANT?')                   0059
      READ(NCR,2) ANAME                                                      0060
      IF(ANAME.NE.'YES'.AND.ANAME.NE.'NO') GO TO 901                         0061
      IF(ANAME.EQ.'NO') GO TO 503                                            0062
      IF(MCAN.EQ.1) WRITE(NTP,5006)                                          0063
 5006 FORMAT(/T2,'ENTER CONSTANT RELATIVE VOLATILITY VALUE.')                0064
C     CONSTANT RELATIVE VOLATILITY **********************************************
      READ(NCR,1) ALFCON                                                     0065
      MALF=2                                                                 0066
```

```
C     SIDE STREAMS.
      IF(MCAN.EQ.1) WRITE(NTP,2010)                                      0132
 2010 FORMAT(/T2,'ENTER SIDE STREAM COMPOSITIONS.')                      0133
      READ(NCR,1) (ZS(I),I=1,NS)                                         0134
      IF(MCAN.EQ.1) WRITE(NTP,2011)                                      0135
 2011 FORMAT(/T2,'ENTER SIDE STREAM Q VALUES.')                          0136
      READ(NCR,1) (QS(I),I=1,NS)                                         0137
      IF(MCAN.EQ.1) WRITE(NTP,2012)                                      0138
 2012 FORMAT(/T2,'ENTER SIDE STREAM WITHDRAWAL RATES.')                  0139
      READ(NCR,1) (STOTS(I),I=1,NS)                                      0140
      GO TO 212                                                          0141
  211 CONTINUE                                                           0142
C     SIMPLE COLUMNS.
      NF=1                                                               0143
      NS=0                                                               0144
      IF(MCAN.EQ.1) WRITE(NTP,2013)                                      0145
 2013 FORMAT(/T2,'ENTER FEED COMPOSITION AND FEED RATE.')                0146
      READ(NCR,1) Z,FTOT                                                 0147
C     FEED Q VALUE MUST BE SPECIFIED FOR MCCABE-THIELE DESIGNS.
      IF(MCAN.EQ.1.AND.NOP.EQ.1) WRITE(NTP,2023)                         0148
 2023 FORMAT(T2,'ENTER FEED Q VALUE.')                                   0149
      IF(NOP.EQ.1) READ(NCR,1) Q                                         0150
C     TEMPERATURE OF FEED REQUIRED FOR PONCHON-SAVARIT DESIGNS ONLY.
      IF(MCAN.EQ.1.AND.NOP.GT.1) WRITE(NTP,2014)                         0151
 2014 FORMAT(T2,'ENTER TEMPERATURE OF FEED ...(DEG C)')                  0152
      IF(NOP.GT.1) READ(NCR,1) TF                                        0153
  212 CONTINUE                                                           0154
      IF(NOP.GT.1) GO TO 213                                             0155
C     OPEN STEAM OR PARTIAL REBOILER **********************************
      IF(MCAN.EQ.1) WRITE(NTP,2015)                                      0156
 2015 FORMAT(/T2,'IF THE HEAVY COMPONENT IS WATER AND OPEN STEAM',/T2,   0157
     1'OPERATION IS REQUIRED,ENTER 1 ...IF NOT ENTER 0.')                0158
      READ(NCR,1) NOS                                                    0159
      IF(NOS.GT.1) GO TO 901                                             0160
  213 CONTINUE                                                           0161
```

```
      IF(NOP.EQ.2) GO TO 214                                            0162
C     EFFICIENCY FACTOR ***************************************************
      IF(MCAN.EQ.1) WRITE(NTP,2016)                                     0163
 2016 FORMAT(/T2,'ENTER MURPHREE STAGE EFFICIENCY FACTOR.',/T2,         0164
     1'(=1.0 IF PERFECT TRAYS ARE ASSUMED)')                            0165
      READ(NCR,1) EFF                                                   0166
  214 CONTINUE                                                          0167
      IF(NOP.EQ.1) GO TO 999                                            0168
C     ENTHALPY DATA *******************************************************
C     REQUIRED FOR PONCHON-SAVARIT DESIGNS.
      WRITE(NTP,3010)                                                   0169
 3010 FORMAT(/T2,'ADD ENTHALPY DATA ELEMENT. (SEE MANUAL SECTION 2)')   0170
C     ENTHALPY FUNCTION COEFFICIENTS AND ASSOCIATED POWERS OF TEN.
      READ(NCR,1) (ACP(I),BCP(I),CCP(I),DCP(I),I=1,2)                   0171
      READ(NCR,1) NBCP,NCCP,NDCP                                        0172
      DO 31 I=1,2                                                       0173
      BCP(I)=BCP(I)*10.**NBCP                                           0174
      CCP(I)=CCP(I)*10.**NCCP                                           0175
      DCP(I)=DCP(I)*10.**NDCP                                           0176
   31 CONTINUE                                                          0177
C     CONVERSION FACTORS REQUIRED FOR A CONSISTENT SET OF UNITS.
      READ(NCR,1) CPCONV,LHCONV,HSCONV,HEVABS                           0178
C     BASE TEMPERATURE OF ENTHALPY CALCULATIONS AND ABSOLUTE SCALE
C     CONVERSION TEMPERATURE.
      READ(NCR,1) TBASE,TCONV                                           0179
C     LATENT HEAT,BOILING POINT AND CRITICAL-TEMPERATURE DATA.
      READ(NCR,1) (LH(I),TBP(I),TC(I),I=1,2)                            0180
      TG=(TBP(1)+TBP(2))/2.                                             0181
C     MOLECULAR WEIGHTS: CP-,LH-,HSMW REQUIRED IF RESPECTIVE DATA IS
C     SUPPLIED PER WEIGHT RATHER THAN PER MOLE. (SEE MANUAL)
      READ(NCR,1) (MW(I),CPMW(I),LHMW(I),HSMW(I),I=1,2)                 0182
C     HEAT OF MIXING DATA.
      READ(NCR,1) NHS                                                   0183
      IF(NHS.GT.0) READ(NCR,1) (XHS(I),HSOLN(I),I=1,NHS)                0184
      CALL FLASHB                                                       0185
      GO TO 999                                                         0186
```

```
C      ERROR MESSAGES **********************************************************
  901 CONTINUE                                                                  0187
      WRITE(NTP,9001)                                                           0188
 9001 FORMAT(/T2,'ERROR IN DATA INPUT...ERROR TERMINATION.')                    0189
      MALF=1000                                                                 0190
      GO TO 999                                                                 0191
  902 CONTINUE                                                                  0192
      WRITE(NTP,9002)                                                           0193
 9002 FORMAT(/T2,'DIMENSION SPECIFICATION EXCEEDED.')                           0194
      MALF=1000                                                                 0195
  999 CONTINUE                                                                  0196
      RETURN                                                                    0197
      END                                                                       0198

                                                                  SUBROUTINE READBD
```

```
      SUBROUTINE CONDEN(NCAN)                                          0001
C     CALCULATION OF THE COMPOSITIONS OF STREAMS LEAVING THE CONDENSER.
C     ****************************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     ****************************************
      REAL LH,LHCONV,LHMV,LIQR,LIQS,LRWT,LSWT,MW                       0002
    1 FORMAT( )                                                        0003
    2 FORMAT(A3)                                                       0004
      GO TO (100,200),NCOND                                            0005
  100 CONTINUE                                                         0006
C     TOTAL CONDENSER
      XC=XD                                                            0007
      YC=XD                                                            0008
      GO TO 999                                                        0009
  200 CONTINUE                                                         0010
C     PARTIAL CONDENSER
      IF(NOP.GT.1) GO TO 210                                           0011
      IF(NCAN.EQ.1) WRITE(NTP,1010)                                    0012
 1010 FORMAT(/T2,'ENTER FRACTION OF TOTAL DISTILLATE TO BE WITHDRAWN AS'0013
     1,/T2,'VAPOUR AND LIQUID RESPECTIVELY. IF ALL VAPOUR ENTER 1.,0.')0014
      READ(NCR,1) DV,DL                                                0015
      IF(DL.GT.0.00001) GO TO 211                                      0016
  210 CONTINUE                                                         0017
C     ALL VAPOUR DISTILLATE
      YC=XD                                                            0018
      CALL XYALFA(ALFA,ALFCON,ALPHA,MALF,NDATA,YALF,YC)                0019
      XC=YC/(ALPHA-YC*(ALPHA-1.))                                      0020
      GO TO 999                                                        0021
  211 CONTINUE                                                         0022
C     MIXED VAPOUR AND LIQUID DISTILLATE
      IF(NCAN.EQ.1) WRITE(NTP,2011)                                    0023
 2011 FORMAT(/T2,'IS XD AN OVERALL DISTILLATE COMPOSITION?')           0024
      READ(NCR,2) XONAME                                               0025
      IF(XONAME.EQ.'YES') GO TO 230                                    0026
      IF(NCAN.EQ.1) WRITE(NTP,2012)                                    0027
 2012 FORMAT(/T2,'IS XD THE COMPOSITION OF THE VAPOUR OR LIQUID FRACTION0028
     1.',/T2,'ENTER VAP OR LIQ.')                                      0029
      READ(NCR,2) XLNAME                                               0030
      IF(XLNAME.EQ.'LIQ') GO TO 220                                    0031
```

```
C       XD GIVEN AS COMPOSITION OF VAPOUR FRACTION
        YCC=XD                                                          0032
        CALL XYALFA(ALFA,ALFCON,ALF,MALF,NDATA,YALF,YCC)                0033
        XDC=YCC/(ALF-YCC*(ALF-1.))                                      0034
        CDLDV=DV/DL                                                     0035
        XD=(CDLDV*YCC+XDC)/(1.+CDLDV)                                   0036
        XC=XDC                                                          0037
        YC=YCC                                                          0038
        GO TO 999                                                       0039
  220 CONTINUE                                                          0040
C       XD GIVEN AS COMPOSITION OF LIQUID FRACTION
        XDC=XD                                                          0041
        CALL XYALFA(ALFA,ALFCON,ALF,MALF,NDATA,XALF,XDC)                0042
        YCC=XDC*ALF/(1.+XDC*(ALF-1.))                                   0043
        CDLDV=DV/DL                                                     0044
        XD=(CDLDV*YCC+XDC)/(1.+CDLDV)                                   0045
        XC=XDC                                                          0046
        YC=YCC                                                          0047
        GO TO 999                                                       0048
  230 CONTINUE                                                          0049
C       XD GIVEN AS OVERALL COMPOSITION VALUE
C       MIXED VAPOUR AND LIQUID DISTILLATE
        CDLDV=DV/DL                                                     0050
        DELX=0.001                                                      0051
        MM=2                                                            0052
        X1=XD-DELX                                                      0053
  231 CONTINUE                                                          0054
        CALL XYALFA(ALFA,ALFCON,ALF,MALF,NDATA,XALF,X1)                 0055
        YC=X1*ALF/(1.+X1*(ALF-1.))                                      0056
        FACT=(CDLDV*YC+X1)/(1.+CDLDV)                                   0057
        IF(ABS(FACT-XD).LT.0.00001) GO TO 232                           0058
        NCALL=1                                                         0059
        CALL SEARCH(FACT,XD,X1,DELX,MM,NCALL)                           0060
        GO TO 231                                                       0061
  232 CONTINUE                                                          0062
        XC=X1                                                           0063
  999 CONTINUE                                                          0064
        RETURN                                                          0065
        END                                                             0066
```

```
      SUBROUTINE FLASHB                                                 0001
C     FLASH CALCULATION FOR BINARY SYSTEMS THAT INCLUDE ENTHALPY
C     CONSIDERATIONS.
C     **********************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     **********************************
      REAL LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                        0002
      TDPT=TG                                                           0003
      TBPT=TG                                                           0004
      CALL BINDPT(A,B,C,NBDC,NDATA,PT,PI,TALF,TBP,TDPT,YALF,Z)          0005
      CALL BINBPT(A,B,C,NBDC,NDATA,PT,PI,TALF,TBP,TBPT,XALF,Z)          0006
      IF(TF.LT.TBPT.OR.TF.GT.TDPT) GO TO 120                            0007
      CALL EQUIL(NDATA,TALF,TBP,TF,XALF,YALF,X1,Y1)                     0008
      Q=(Y1-Z)/(Y1-X1)                                                  0009
      GO TO 999                                                         0010
  120 CONTINUE                                                          0011
      CALL HTCAP(TDPT,TDPT,TBPT,Z,Z,HV,HL)                              0012
      CALL HTCAP(TDPT,TF,TF,Z,Z,HVF,HLF)                                0013
      IF(TF.LT.TBPT) HF=HLF                                             0014
      IF(TF.GT.TDPT) HF=HVF                                             0015
      Q=(HV-HF)/(HV-HL)                                                 0016
  999 CONTINUE                                                          0017
      WRITE(NTP,4000) Q,TF,TDPT,TBPT                                    0018
 4000 FORMAT(/T2,'FEED Q VALUE AND TEMPERATURE DATA:',/T2,'Q:',F8.4,    0019
     1 15X,'FEED TEMP:',F8.2,1X,'DEG C',/T2,'DEW PT:',F8.2,1X,'DEG C',4X,0020
     2'BUBBLE PT:',F8.2,1X,'DEG C',/)                                   0021
      RETURN                                                            0022
      END                                                               0023
```

```
      SUBROUTINE EQUIL(NDATA,TALF,TBP,TF,XALF,YALF,X1,Y1)                    0001
C     'EQUIL' SEARCHES FOR EQUILIBRIUM DATA POINTS CORRESPONDING TO
C     TEMPERATURE TF,USING SUPPLIED OR CALCULATED X-Y-T DATA. IF TF NE TALF
C     LINEAR INTERPOLATION IS USED TO FIND VALUES FOR XALF AND YALF.
      DIMENSION TALF(1),TBP(1),XALF(1),YALF(1)                               0002
      IF(TF.LT.TALF(1)) GO TO 101                                            0003
      TFACT=(TBP(2)-TF)/(TBP(2)-TALF(1))                                     0004
      X1=0.0+TFACT*XALF(1)                                                   0005
      Y1=TFACT*YALF(1)                                                       0006
      GO TO 999                                                              0007
  101 CONTINUE                                                               0008
      IF(TF.LT.TALF(NDATA)) GO TO 102                                        0009
      DO 10 I=1,NDATA                                                        0010
      DIFF=TALF(I)-TF                                                        0011
      IF(DIFF.LE.0.0) GO TO 110                                              0012
   10 CONTINUE                                                               0013
      GO TO 102                                                              0014
  110 CONTINUE                                                               0015
      TFACT=(TALF(I-1)-TF)/(TALF(I-1)-TALF(I))                               0016
      X1=XALF(I-1)+TFACT*(XALF(I)-XALF(I-1))                                 0017
      Y1=YALF(I-1)+TFACT*(YALF(I)-YALF(I-1))                                 0018
      GO TO 999                                                              0019
  102 CONTINUE                                                               0020
      TFACT=(TALF(NDATA)-TF)/(TALF(NDATA)-TBP(1))                            0021
      X1=XALF(NDATA)+TFACT*(1.0-XALF(NDATA))                                 0022
      Y1=YALF(NDATA)+TFACT*(1.0-YALF(NDATA))                                 0023
  999 CONTINUE                                                               0024
      RETURN                                                                 0025
      END                                                                    0026

                                                          SUBROUTINE EQUIL
```

```
      SUBROUTINE MULTFS(NN1)                                              0001
C     ROUTINE FOR HANDLING BINARY PROBLEMS WITH MORE THAN ONE FEED
C     AND/OR SIDE STREAM.
C     THE ROUTINE: (1) SORTS THE FEED AND SIDE STREAMS INTO AN ORDER OF
C                      INCREASING CONCENTRATION.
C                  (2) COMPUTES BOTTOMS FLOWRATE AND COMPOSITION FROM AN
C                      OVERALL MASS BALANCE.
C                  (3) COMPUTES LIQUID AND VAPOUR FLOWRATES IN EACH
C                      SECTION OF COLUMN.
C     ************************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     ************************************
      REAL LH,LHCONV,LHNW,LIQR,LIQS,LRWT,LSWT,MW                          0002
C     OVERALL MASS BALANCE.
      SUMF=0.                                                             0003
      SUMFZ=0.                                                            0004
      SUMFQ=0.                                                            0005
      SUMS=0.                                                             0006
      SUMSZ=0.                                                            0007
      SUMSQ=0.                                                            0008
C     'DO' LOOP 11 COMPUTES OVERALL FEED FLOW,COMPOSITION AND Q VALUE.
      DO 11 I=1,NF                                                        0009
      SUMF=SUMF+FTOTS(I)                                                  0010
      SUMFZ=SUMFZ+FTOTS(I)*ZF(I)                                          0011
      SUMFQ=SUMFQ+FTOTS(I)*QF(I)                                          0012
   11 CONTINUE                                                            0013
      IF(NS.EQ.0) GO TO 101                                               0014
C     'DO' LOOP 12 COMPUTES OVERALL SIDE FLOW,COMPOSITION AND Q VALUE.
      DO 12 I=1,NS                                                        0015
      SUMS=SUMS+STOTS(I)                                                  0016
      SUMSZ=SUMSZ+STOTS(I)*ZS(I)                                          0017
      SUMSQ=SUMSQ+STOTS(I)*QS(I)                                          0018
   12 CONTINUE                                                            0019
```

```
  101 CONTINUE                                                          0020
C     RESULTANT OVERALL FEED Q VALUE AND COMPOSITION.
      QMIX=(SUMFQ-SUMSQ)/(SUMF-SUMS)                                    0021
      ZMIX=(SUMFZ-SUMSZ)/(SUMF-SUMS)                                    0022
      IF(NOS.EQ.1) GO TO 102                                            0023
C     BOTTOMS FLOWRATE AND COMPOSITION FOR PARTIAL REBOILER OPERATION.
      BTOT=SUMF-SUMS-DTOT                                               0024
      XB=(SUMFZ-SUMSZ-DTOT*XD)/BTOT                                     0025
      IF(BTOT.LT.0.0.OR.XB.LT.0.0) GO TO 902                            0026
      YB=XB                                                             0027
      GO TO 103                                                         0028
  102 CONTINUE                                                          0029
C     BOTTOMS FLOWRATE AND COMPOSITION FOR OPEN STEAM OPERATION.
      YB=0.0                                                            0030
      GOP=DTOT*(R+1.)+(QMIX-1.)*(SUMF-SUMS)                             0031
      BTOT=SUMF-SUMS-DTOT+GOP                                           0032
      XB=(SUMFZ-SUMSZ-DTOT*XD)/BTOT                                     0033
      IF(BTOT.LT.0.0.OR.XB.LT.0.0) GO TO 902                            0034
  103 CONTINUE                                                          0035
      YD=XD                                                             0036
C     POINT OF INTERSECTION OF TOP AND BOTTOM OVERALL OPERATING LINES.
      XQM=((R+1.)*ZMIX+(QMIX-1.)*XD)/(R+QMIX)                           0037
      YQM=(R*ZMIX+QMIX*XD)/(R+QMIX)                                     0038
      SLOPB=(YQM-YB)/(XQM-XB)                                           0039
C     FLOWRATE IN BOTTOM SECTION OF COLUMN : PARTIAL REBOILER OPERATION.
      AV(1)=BTOT/(SLOPB-1.)                                             0040
      AL(1)=BTOT+AV(1)                                                  0041
C     FLOWRATE IN BOTTOM SECTION OF COLUMN : OPEN STEAM OPERATION.
      IF(NOS.EQ.1) AV(1)=GOP                                            0042
      IF(NOS.EQ.1) AL(1)=BTOT                                           0043
      CB=YB-SLOPB*XB                                                    0044
```

```
      NN=NF+NS+1                                                        0045
C     'DO' LOOP 21 SORTS FEED AND SIDE STREAMS ON BASIS OF COMPOSITION.
      DO 21 I=NN,2,-1                                                   0046
      ZMIN=XB                                                           0047
      DO 22 K=1,NS                                                      0048
      ZMAX=ZS(K)                                                        0049
      IF(ZMIN-ZMAX) 201,901,202                                         0050
  201 ZN=ZMAX                                                           0051
      ZMIN=ZMAX                                                         0052
      NI=K                                                              0053
      NNN=1                                                             0054
  202 CONTINUE                                                          0055
   22 CONTINUE                                                          0056
      DO 23 L=1,NF                                                      0057
      ZMAX=ZF(L)                                                        0058
      IF(ZMIN-ZMAX) 203,901,204                                         0059
  203 ZN=ZMAX                                                           0060
      ZMIN=ZMAX                                                         0061
      NI=L                                                              0062
      NNN=3                                                             0063
  204 CONTINUE                                                          0064
   23 CONTINUE                                                          0065
      IF(NNN-2) 205,206,206                                             0066
C     COMPOSITION,FLOWRATE AND Q VALUE OF STREAMS IN ORDER OF
C     DECREASING COMPOSITION FROM THE TOP OF THE COLUMN.
  205 ZFS(I)=ZN                                                         0067
      FLOW(I)=-STOTS(NI)                                                0068
      QFS(I)=QS(NI)                                                     0069
      ZS1(NI)=ZS(NI)                                                    0070
      ZS(NI)=-1.                                                        0071
      GO TO 21                                                          0072
  206 ZFS(I)=ZN                                                         0073
      FLOW(I)=FTOTS(NI)                                                 0074
      QFS(I)=QF(NI)                                                     0075
      ZF1(NI)=ZF(NI)                                                    0076
      ZF(NI)=-1.                                                        0077
   21 CONTINUE                                                          0078
```

```
C       SLOPE AND Y-AXIS INTERCEPT OF OPERATING LINES IN EACH SECTION
C       OF THE COLUMN.
        SLPE(1)=SLOPB                                                          0079
        CFCT(1)=CB                                                             0080
        QX(1)=XB                                                               0081
        QY(1)=YB                                                               0082
        NN1=NN+1                                                               0083
        SLPE(NN1)=1.0                                                          0084
        CFCT(NN1)=0.0                                                          0085
C       'DO' LOOP 24 COMPUTES POINTS OF INTERSECTION OF OPERATING LINES,
C       AND LIQUID AND VAPOUR FLOWRATES WITHIN EACH SECTION.
        DO 24 I=2,NN                                                           0086
        II=I-1                                                                 0087
        QX(I)=(ZFS(I)-CFCT(II)*(1.-QFS(I)))/(SLPE(II)*(1.-QFS(I))+QFS(I))      0088
        QY(I)=SLPE(II)*QX(I)+CFCT(II)                                          0089
        AL(I)=AL(II)-QFS(I)*FLOW(I)                                            0090
        AV(I)=AV(II)+(1.-QFS(I))*FLOW(I)                                       0091
        SLPE(I)=AL(I)/AV(I)                                                    0092
        CFCT(I)=QY(I)-SLPE(I)*QX(I)                                            0093
   24   CONTINUE                                                               0094
        QX(NN1)=XD                                                             0095
        QY(NN1)=YD                                                             0096
        GO TO 999                                                              0097
C       ERROR MESSAGES ***************************************************
  901   CONTINUE                                                               0098
        MALF=1000                                                              0099
        WRITE(NTP,9001)                                                        0100
 9001   FORMAT(/T2,'ERROR...TWO STREAMS HAVE THE SAME COMPOSITION.')           0101
        GO TO 999                                                              0102
  902   CONTINUE                                                               0103
        MALF=1000                                                              0104
        WRITE(NTP,9002)                                                        0105
 9002   FORMAT(/T2,'ERROR...BTOT OR XB IS LESS THAN 0.0.')                     0106
  999   CONTINUE                                                               0107
        RETURN                                                                 0108
        END                                                                    0109
```

```
      SUBROUTINE SIMPLE                                                 0001
C     ROUTINE FOR DETERMINING OPERATING CONDITIONS FOR BINARY PROBLEMS
C     INVOLVING SIMPLE COLUMNS. (ONE FEED,NO SIDE STREAMS)
C     THE ROUTINE: (1) COMPUTES BOTTOMS FLOWRATE AND COMPOSITION FROM AN
C                      OVERALL MASS BALANCE.
C                  (2) DETERMINES OPERATING LINE EQUATIONS.
C                  (3) COMPUTES LIQUID AND VAPOUR FLOWRATES IN EACH
C                      SECTION OF COLUMN.
C     ************************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     ************************************
      REAL LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                        0002
      ZFS(2)=Z                                                          0003
      QFS(2)=Q                                                          0004
      FLOW(2)=FTOT                                                      0005
C     POINT OF INTERSECTION OF OPERATING LINES.
      XQM=((R+1.)*Z+(Q-1.)*XD)/(R+Q)                                    0006
      YQM=(R*Z+Q*XD)/(R+Q)                                              0007
      IF(NOS.EQ.1) GO TO 110                                            0008
C     BOTTOMS FLOWRATE AND COMPOSITION FOR PARTIAL REBOILER OPERATION.
      BTOT=FTOT-DTOT                                                    0009
      XB=(FTOT*Z-DTOT*XD)/BTOT                                          0010
      IF(XB.LT.0.0.OR.XB.GT.Z) GO TO 901                                0011
      YB=XB                                                             0012
      GO TO 120                                                         0013
  110 CONTINUE                                                          0014
C     BOTTOMS FLOWRATE AND COMPOSITION FOR OPEN STEAM OPERATION.
      YB=0.0                                                            0015
      GOP=DTOT*(R+1.)+(Q-1.)*FTOT                                       0016
      BTOT=FTOT-DTOT+GOP                                                0017
      XB=(FTOT*Z-DTOT*XD)/BTOT                                          0018
      IF(XB.LT.0.0.OR.XB.GT.Z) GO TO 901                                0019
  120 CONTINUE                                                          0020
      YD=XD                                                             0021
```

```
C     SLOPES AND Y-AXIS INTERCEPTS OF OPERATING LINES.
      SLPE(1)=(YQM-YB)/(XQM-XB)                                          0022
      CFCT(1)=YQM-SLPE(1)*XQM                                            0023
      SLPE(2)=R/(R+1.)                                                   0024
      CFCT(2)=XD/(R+1.)                                                  0025
C     VAPOUR AND LIQUID FLOWRATES WITHIN THE COLUMN.
      AV(1)=BTOT/(SLPE(1)-1.)                                            0026
      AL(1)=BTOT+AV(1)                                                   0027
      IF(NOS.EQ.1) AV(1)=GOP                                             0028
      IF(NOS.EQ.1) AL(1)=BTOT                                            0029
      AL(2)=DTOT*R                                                       0030
      AV(2)=DTOT+AL(2)                                                   0031
      SLPE(3)=1.0                                                        0032
      CFCT(3)=0.0                                                        0033
C     POINTS OF INTERSECTION OF OPERATING LINES WITH DIAGONAL AND EACH OTHER.
      QX(1)=XB                                                           0034
      QY(1)=YB                                                           0035
      QX(2)=XQM                                                          0036
      QY(2)=YQM                                                          0037
      QX(3)=XD                                                           0038
      QY(3)=YD                                                           0039
      NN=2                                                               0040
      GO TO 999                                                          0041
C     ERROR MESSAGES **************************************************
  901 CONTINUE                                                           0042
      WRITE(NTP,9001)                                                    0043
 9001 FORMAT(/T2,'ERROR...BTOT OR XB IS LESS THAN 0.0',//T10,'OR XB IS G 0044
     1REATER THAN 2.')                                                   0045
      MALF=1000                                                          0046
  999 CONTINUE                                                           0047
      RETURN                                                             0048
      END                                                                0049
```

SUBROUTINE SIMPLE

```
      SUBROUTINE PONSAV(NPTS,MPTS,NFIN,MFIN)                            0001
C     ROUTINE FOR CONSTRUCTION OF PONCHON-SAVARIT OPERATING DIAGRAM.
C     THE ROUTINE: (1) COMPUTES BOTTOMS FLOWRATE AND COMPOSITION FROM AN
C                      OVERALL MASS BALANCE.
C                  (2) CONSTRUCTS ENTHALPY- CONCENTRATION DIAGRAM.
C                  (3) COMPUTES MINIMUM REFLUX RATIO.
C                  (4) DEFINES OPERATING LINES ON X,Y DIAGRAM.
C                  (5) COMPUTES VAPOUR AND LIQUID MASS AND MOLAR FLOWS
C                      THROUGH COLUMN.
C                  (6) DETERMINES CONDENSER AND REBOILER HEAT DUTIES.
C     ************************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     ************************************
      REAL LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                        0002
      DATA CONFAC,ENCONV/0.00027778,2.326/                              0003
C     CALCULATION OF BOTTOMS FLOWRATE AND COMPOSITION.
      BTOT=FTOT-DTOT                                                    0004
      XB=(FTOT*Z-DTOT*XD)/BTOT                                          0005
      IF(XB.LT.0.0.OR.XB.GT.Z) GO TO 901                                0006
      TDG=TG                                                            0007
      DELY=0.05                                                         0008
      DELX=0.05                                                         0009
      YH(1)=0.0                                                         0010
      XH(1)=0.0                                                         0011
      NENTH=0                                                           0012
C     'DO' LOOP 10 COMPUTES DATA REQUIRED TO CONSTRUCT ENTHALPY-
C     CONCENTRATION DIAGRAM IN PONCHON-SAVARIT DESIGNS.
      DO 10 K=1,100                                                     0013
      XG=XH(K)                                                          0014
      YG=YH(K)                                                          0015
      IF(YG.GT.1.01) GO TO 100                                          0016
      NENTH=NENTH+1                                                     0017
      TA=TDG                                                            0018
C     'BINDPT' CALCULATES DEW POINTS.
      CALL BINDPT(A,B,C,NBDC,NDATA,PT,PA,TALF,TBP,TA,YALF,YG)           0019
      TDU=TA                                                            0020
```

```
      TA=TDG                                                              0021
C     'BINBPT' CALCULATES BUBBLE POINTS.
      CALL BINBPT(A,B,C,NBDC,NDATA,PT,PA,TALF,TBP,TA,XALF,XG)             0022
      TBU=TA                                                              0023
C     'HTCAP' COMPUTES VAPOUR AND LIQUID ENTHALPY.
      CALL HTCAP(TDU,TDU,TBU,YG,XG,HV,HL)                                 0024
      HVAP(K)=HV*ENCONV                                                   0025
      HLIQ(K)=HL*ENCONV                                                   0026
      YH(K+1)=YH(K)+DELY                                                  0027
      XH(K+1)=XH(K)+DELX                                                  0028
   10 CONTINUE                                                            0029
  100 CONTINUE                                                            0030
C     COMPUTATION OF FEED ENTHALPY.
      TDF=TG                                                              0031
      TBF=TG                                                              0032
      CALL BINDPT(A,B,C,NBDC,NDATA,PT,PA,TALF,TBP,TDF,YALF,Z)             0033
      CALL BINBPT(A,B,C,NBDC,NDATA,PT,PA,TALF,TBP,TBF,XALF,Z)             0034
      CALL HTCAP(TDF,TDF,TBF,Z,Z,HVF,HLF)                                 0035
      HF=(HVF-Q*(HVF-HLF))*ENCONV                                         0036
C     'RMIN' COMPUTES THE MINIMUM REFLUX RATIO.
      CALL RMIN(HLD,HVD)                                                  0037
      IF(MALF.GT.999) GO TO 999                                           0038
C     CHECK IF SPECIFIED REFLUX RATIO IS ACCEPTABLE. IF NOT USE 1.5*RM
      IF(RM.GT.R) GO TO 902                                               0039
      GO TO 190                                                           0040
  199 R=1.5*RM                                                            0041
  190 CONTINUE                                                            0042
C     SEARCH FOR DISTILLATE ENTHALPY DEPENDENT ON CONDENSER TYPE.
      IF(NCOND.EQ.1) CALL HFIND(XD,HD,HLIQ,NENTH,XH)                      0043
      IF(NCOND.EQ.2) CALL HFIND(XD,HD,HVAP,NENTH,YH)                      0044
C     CALCULATION OF REBOILER AND CONDENSER HEAT DUTY,AND POSITION OF
C     'DEL' POINTS ON ENTHALPY-CONCENTRATION DIAGRAM.
      Q1DASH=R*(HVD-HLD)+HVD                                              0045
      QC=(Q1DASH-HD)*DTOT                                                 0046
      Q2DASH=(FTOT*HF-DTOT*Q1DASH)/BTOT                                   0047
C     'HFIND' RETURNS BOTTOMS ENTHALPY HB (AT CONC XB).
      CALL HFIND(XB,HB,HLIQ,NENTH,XH)                                     0048
      QB=(HB-Q2DASH)*BTOT                                                 0049
```

```
      DELD=Q1DASH                                                       0050
      DELB=Q2DASH                                                       0051
      SLOPE=(DELD-HF)/(XD-Z)                                            0052
      CFACT=DELD-XD*SLOPE                                               0053
      DELXF=0.01                                                        0054
      MN=1                                                              0055
      XG=Z+DELXF                                                        0056
C     'GO TO' LOOPS 201 AND 202 SEARCH FOR THE POINT OF INTERSECTION OF
C     THE OPERATING LINES. (XCR,YCR)
  201 CONTINUE                                                          0057
      CALL HFIND(XG,HLC,HLIQ,NENTH,XH)                                  0058
      HG=SLOPE*XG+CFACT                                                 0059
      IF(ABS(HLC-HG).LT.10.) GO TO 210                                  0060
      NCALL=2                                                           0061
      CALL SEARCH(HLC,HG,XG,DELXF,MN,NCALL)                             0062
      GO TO 201                                                         0063
  210 CONTINUE                                                          0064
      XCR=XG                                                            0065
      DELYF=0.01                                                        0066
      MN=1                                                              0067
      YG=Z+DELYF                                                        0068
  202 CONTINUE                                                          0069
      CALL HFIND(YG,HVC,HVAP,NENTH,YH)                                  0070
      HG=SLOPE*YG+CFACT                                                 0071
      IF(ABS(HVC-HG).LT.10.) GO TO 220                                  0072
      NCALL=2                                                           0073
      CALL SEARCH(HVC,HG,YG,DELYF,MN,NCALL)                             0074
      GO TO 202                                                         0075
  220 CONTINUE                                                          0076
      YCR=YG                                                            0077
C     'OPLINE' COMPUTES COORDINATES OF POINTS ON THE OPERATING LINES,
C     USING THE PONCHON-SAVARIT TECHNIQUE.
      CALL OPLINE(NPTS,MPTS)                                            0078
      IF(MALF.GT.999) GO TO 999                                         0079
C     'FLOWS' COMPUTES LIQUID AND VAPOUR FLOWRATES WITHIN THE COLUMN.
      CALL FLOWS(NPTS,MPTS,NFIN,MFIN)                                   0080
      IF(MALF.GT.999) GO TO 999                                         0081
```

```
C       CONVERSION OF HEAT DUTIES AND ENTHALPY TO S.I. UNITS.
C        (QF(1) IS USED TO STORE FEED ENTHALPY)
        QC=QC*CONFAC                                                    0082
        QB=QB*CONFAC                                                    0083
        QF(1)=HF                                                        0084
        IF(NOP.EQ.2) GO TO 999                                          0085
        YB=XB                                                           0086
        YD=XD                                                           0087
        ZFS(2)=Z                                                        0088
        QFS(2)=Q                                                        0089
        FLOW(2)=FTOT                                                    0090
        NN=2                                                            0091
C       POINTS OF INTERSECTION OF OPERATING LINES WITH DIAGONAL AND EACH OTHER.
        QX(1)=XB                                                        0092
        QY(1)=YB                                                        0093
        QX(2)=XCR                                                       0094
        QY(2)=YCR                                                       0095
        QX(3)=XD                                                        0096
        QY(3)=YD                                                        0097
        GO TO 999                                                       0098
C       ERROR MESSAGES **************************************************
  901 CONTINUE                                                          0099
        WRITE(NTP,9001)                                                 0100
 9001 FORMAT(/T2,'ERROR...BTOT OR XB IS LESS THAN 0.0',/T10,'OR XB IS G 0101
     1REATER THAN Z.')                                                  0102
        MALF=1000                                                       0103
        GO TO 999                                                       0104
  902 CONTINUE                                                          0105
        WRITE(NTP,9002) RM,R                                            0106
 9002 FORMAT(/T2,'SPECIFIED REFLUX RATIO IS LESS THAN THE CALCULATED MIN 0107
     1IMUM.',/T4,'RMIN=',F6.2,2X,'R SPEC=',F6.2,/T2,                    0108
     2'A VALUE OF 1.5*RMIN WILL BE USED.')                              0109
        GO TO 199                                                       0110
  999 CONTINUE                                                          0111
        RETURN                                                          0112
        END                                                             0113
```

```
      SUBROUTINE HTCAP(TDPT,T2D,T2B,YS,XS,HV,HL)                        0001
C     ROUTINE FOR COMPUTING VAPOUR AND LIQUID ENTHALPY FOR BINARY SYSTEMS.
C     ************************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     ************************************
      DIMENSION CP(2),LHT(2)                                            0002
      REAL LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                        0003
      REAL LHT                                                          0004
      T=TBASE+TCONV                                                     0005
      TD=T2D+TCONV                                                      0006
      TB=T2B+TCONV                                                      0007
C     VAPOUR ENTHALPY CALCULATIONS.
C     'DO' LOOP 10 COMPUTES HEAT CAPACITY OF EACH COMPONENT AT THE
C     BUBBLE POINT,RELATIVE TO THE BASE TEMPERATURE.
      DO 10 I=1,2                                                       0008
      AFACT=ACP(I)*(TD-T)                                               0009
      BFACT=BCP(I)*(TD**2,-T**2.)/2.                                    0010
      CFACT=CCP(I)*(TD**3,-T**3.)/3.                                    0011
      DFACT=DCP(I)*(TD**4,-T**4.)/4.                                    0012
      CP(I)=AFACT+BFACT+CFACT+DFACT                                     0013
   10 CONTINUE                                                          0014
      Y1=CP(1)*YS*CPMW(1)*CPCONV                                        0015
      Y2=CP(2)*(1.-YS)*CPMW(2)*CPCONV                                   0016
C     LATENT HEAT CALCULATIONS.
      DO 11 I=1,2                                                       0017
      TFACT=(TC(I)-TDPT)/(TC(I)-TBP(I))                                 0018
      LHT(I)=LH(I)*(TFACT**0.38)*LHMW(I)*LHCONV                         0019
   11 CONTINUE                                                          0020
C     TOTAL VAPOUR ENTHALPY.
      HV=Y1+Y2+YS*LHT(1)+(1.-YS)*LHT(2)                                 0021
```

```
C       LIQUID ENTHALPY CALCULATIONS.
C       'DO' LOOP 20 COMPUTES HEAT CAPACITY OF EACH COMPONENT AT THE
C       DEW POINT,RELATIVE TO THE BASE TEMPERATURE.
        DO 20 I=1,2                                                     0022
        AFACT=ACP(I)*(TB-T)                                             0023
        BFACT=BCP(I)*(TB**2.-T**2.)/2.                                  0024
        CFACT=CCP(I)*(TB**3.-T**3.)/3.                                  0025
        DFACT=DCP(I)*(TB**4.-T**4.)/4.                                  0026
        CP(I)=AFACT+BFACT+CFACT+DFACT                                   0027
   20   CONTINUE                                                        0028
C       HEAT OF MIXING INCLUDED IF SPECIFIED.
        IF(NHS.GT.0) CALL HTSOLN(HS,XS,HEVABS,HSCONV,HSMW,HSOLN,NHS,XHS)  0029
        X1=CP(1)*CPMW(1)*CPCONV                                         0030
        X2=CP(2)*CPMW(2)*CPCONV                                         0031
        IF(NHS.EQ.0) HS=0.0                                             0032
C       TOTAL LIQUID ENTHALPY.
        HL=X1*XS+X2*(1.-XS)+HS                                          0033
        RETURN                                                          0034
        END                                                             0035
```

```
      SUBROUTINE HTSOLN(H,XA,HEVABS,HSCONV,HSMW,HSOLN,NHS,XHS)                0001
C     ROUTINE FOR COMPUTING HEATS OF SOLUTION FOR BINARY MIXTURES.
C     LINEAR INTERPOLATION IS USED BETWEEN DATA POINTS.
      DIMENSION HSMW(1),HSOLN(1),XHS(1)                                       0002
      IF(XA.GT.XHS(1)) GO TO 100                                              0003
      H=HSOLN(1)*HSMW(1)*HSCONV*HEVABS*XA                                     0004
      GO TO 999                                                               0005
  100 CONTINUE                                                                0006
      DO 10 I=1,NHS                                                           0007
      DIFF=XA-XHS(I)                                                          0008
      IF(DIFF.LT.0.0) GO TO 110                                               0009
   10 CONTINUE                                                                0010
      GO TO 120                                                               0011
  110 XFACT=(XHS(I)-XA)/(XHS(I)-XHS(I-1))                                     0012
      HFACT=HSCONV*HEVABS*XA                                                  0013
      H=(HSOLN(I)+(HSOLN(I-1)-HSOLN(I))*XFACT)*HSMW(I)*HFACT                  0014
      GO TO 999                                                               0015
  120 H=HSOLN(NHS)*HSMW(1)*HSCONV*HEVABS*XA                                   0016
  999 CONTINUE                                                                0017
      RETURN                                                                  0018
      END                                                                     0019
```

SUBROUTINE HTSOLN

```
      SUBROUTINE HFIND(XA,H,HENTH,NENTH,XYH)                            0001
C     ROUTINE SEARCHES FOR VAPOUR OR LIQUID ENTHALPY VALUE CORRESPONDING
C     TO THE COMPOSITION AND PHASE SUPPLIED.
C     LINEAR INTERPOLATION IS USED BETWEEN ACTUAL DATA POINTS.
      DIMENSION HENTH(1),XYH(1)                                         0002
      IF(XA.GT.XYH(1)) GO TO 100                                        0003
      H=HENTH(1)                                                        0004
      GO TO 999                                                         0005
  100 CONTINUE                                                          0006
      DO 10 I=1,NENTH                                                   0007
      DIFF=XA-XYH(I)                                                    0008
      IF(DIFF.LT.0.0) GO TO 110                                         0009
   10 CONTINUE                                                          0010
      GO TO 120                                                         0011
  110 XFACT=(XYH(I)-XA)/(XYH(I)-XYH(I-1))                               0012
      H=HENTH(I)+(HENTH(I-1)-HENTH(I))*XFACT                            0013
      GO TO 999                                                         0014
  120 H=HENTH(NENTH)                                                    0015
  999 CONTINUE                                                          0016
      RETURN                                                            0017
      END                                                               0018
```

SUBROUTINE HFIND

```
      SUBROUTINE RMIN(HLD,HVD)                                          0001
C     ROUTINE CALCULATES MINIMUM REFLUX RATIO IN PONCHON-SAVARIT
C     DESIGN OF BINARY SYSTEMS.
C     ***********************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     ***********************************
      REAL LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                        0002
      ACC=0.0001*HF                                                     0003
      DELY=0.02                                                         0004
      Y1=XD                                                             0005
      MN=1                                                              0006
      NCOUNT=0                                                          0007
C     'GO TO' LOOP 100 SEARCHES FOR THE TIE LINE THAT PASSES THROUGH
C     POINT (Z,HF) ON THE ENTHALPY-CONCENTRATION DIAGRAM.
  100 CONTINUE                                                          0008
      NCOUNT=NCOUNT+1                                                   0009
      IF(NCOUNT.GT.500) GO TO 902                                       0010
      CALL XYALFA(ALFA,ALFCON,ALF,MALF,NDATA,YALF,Y1)                   0011
      X1=Y1/(ALF-Y1*(ALF-1.))                                           0012
      IF(NCOUNT.GT.1) GO TO 101                                         0013
      IF(X1.LE.Z) GO TO 901                                             0014
  101 CONTINUE                                                          0015
      CALL HFIND(X1,HL1,HLIQ,NENTH,XH)                                  0016
      CALL HFIND(Y1,HV1,HVAP,NENTH,YH)                                  0017
      SLOPE=(HV1-HL1)/(Y1-X1)                                           0018
      CFACT=HL1-SLOPE*X1                                                0019
      HTEST=SLOPE*Z+CFACT                                               0020
      IF(ABS(HTEST-HF).LT.ACC) GO TO 200                                0021
      NCALL=2                                                           0022
      CALL SEARCH(HTEST,HF,Y1,DELY,MN,NCALL)                            0023
      GO TO 100                                                         0024
  200 CONTINUE                                                          0025
```

SUBROUTINE RMIN

```
C     X,Y COORDINATES OF TIE LINE REPRESENT A POINT OF INTERSECTION OF
C     THE OPERATING LINES THAT FALLS ON THE EQUILIBRIUM CURVE,
C     THIS IS A STATE OF MINIMUM REFLUX.
      DELDM=XD*SLOPE+CFACT                                          0026
C     'HFIND' RETURNS ENTHALPY OF REFLUX LIQUID.
      CALL HFIND(XC,HLD,HLIQ,NENTH,XH)                              0027
      IF(NCOND.EQ.2) GO TO 210                                      0028
C     TOTAL CONDENSER OPERATION.
C     'HFIND' RETURNS ENTHALPY OF VAPOUR ENTERING TOTAL CONDENSER.
      CALL HFIND(XD,HVD,HVAP,NENTH,YH)                              0029
      GO TO 220                                                     0030
  210 CONTINUE                                                      0031
C     PARTIAL CONDENSER OPERATION.
      SLPM=(DELDM-HLD)/(XD-XC)                                      0032
      CFM=DELDM-XD*SLPM                                             0033
      DELXM=0.0001                                                  0034
      MN=1                                                          0035
      XI=XC+DELXM                                                   0036
      NCOUNT=0                                                      0037
C     'GO TO' LOOP 211 SEARCHES FOR ENTHALPY OF VAPOUR ENTERING PARTIAL
C     CONDENSER.
  211 CONTINUE                                                      0038
      NCOUNT=NCOUNT+1                                               0039
      IF(NCOUNT.GT.500) GO TO 903                                   0040
      CALL HFIND(XI,HVD,HVAP,NENTH,YH)                              0041
      HVCM=XI*SLPM+CFM                                              0042
      IF(ABS(HVD-HVCM).LT.ACC) GO TO 220                            0043
      NCALL=2                                                       0044
      CALL SEARCH(HVD,HVCM,XI,DELXM,MN,NCALL)                       0045
      GO TO 211                                                     0046
  220 CONTINUE                                                      0047
C     MINIMUM REFLUX RATIO.
      RM=(DELDM-HVD)/(HVD-HLD)                                      0048
      GO TO 999                                                     0049
```

```
C     ERROR MESSAGES ************************************************************
  901 CONTINUE                                                                  0050
      WRITE(NTP,9001)                                                           0051
 9001 FORMAT(/T2,'ERROR IN RMIN. FEED ENTERS ON TOP TRAY.')                     0052
      MALF=1000                                                                 0053
      GO TO 999                                                                 0054
  902 CONTINUE                                                                  0055
      WRITE(NTP,9002)                                                           0056
 9002 FORMAT(/T2,'ERROR IN RMIN ROUTINE.',/T2,'TIE LINE PASSING THROUGH        0057
     1POINT (Z,HF) NOT FOUND.')                                                 0058
      WRITE(NTP,9012) HTEST,HF,Y1,X1,NCOUNT,DELY                                0059
 9012 FORMAT(T2,'HTEST=',E12.8,2X,'HF=',E12.8,2X,'Y1=',F6.4,2X,'X1=',           0060
     1F6.4,/T12,'NO. OF ITERATIONS=',I9,2X,'INCREMENT=',E12.8)                  0061
      MALF=1000                                                                 0062
      GO TO 999                                                                 0063
  903 CONTINUE                                                                  0064
      WRITE(NTP,9003)                                                           0065
 9003 FORMAT(/T2,'ERROR IN RMIN ROUTINE.',/T2,'ENTHALPY OF VAPOUR ENTERI       0066
     1NG PARTIAL CONDENSER NOT FOUND.')                                         0067
      WRITE(NTP,9013) HVD,HVCM,X1,XC,NCOUNT,DELXM                               0068
 9013 FORMAT(T2,'HVD=',E12.8,2X,'HVCM=',E12.8,2X,'X1=',F6.4,2X,'XC=',           0069
     1F6.4,/T12,'NO. OF ITERATIONS=',I9,2X,'INCREMENT=',E12.8)                  0070
      MALF=1000                                                                 0071
  999 CONTINUE                                                                  0072
      RETURN                                                                    0073
      END                                                                       0074
```

```
      SUBROUTINE OPLINE(NPTS,MPTS)                                          0001
C     ROUTINE CALCULATES X,Y POINTS USED TO CONSTRUCT THE OPERATING LINES
C     ON A PONCHON-SAVARIT DIAGRAM.
C     ************************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     ************************************
      REAL LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                            0002
      ACC=0.0001*HF                                                         0003
C     RECTIFYING SECTION.
C     END POINTS OF RECTIFYING SECTION ARE XD AND XCR.
C     THIS INTERVAL IS DIVIDED INTO NINT INCREMENTS OF SIZE H.
      NPTS=NINT+1                                                           0004
      MN=1                                                                  0005
      X1=XD                                                                 0006
      H=(XD-XCR)/NINT                                                       0007
      Y1=X1                                                                 0008
      XROP(NPTS)=X1                                                         0009
      YROP(NPTS)=Y1                                                         0010
C     'GO TO' LOOP 100 USES PONCHON-SAVARIT METHOD TO FIND THE Y-COORDINATE
C     AT EACH VALUE OF X1,FROM XD TO XCR IN STEPS OF -H.
  100 CONTINUE                                                              0011
      NPTS=NPTS-1                                                           0012
      X1=X1-H                                                               0013
      IF(X1.LT.XCR) GO TO 500                                               0014
C     'HFIND' RETURNS LIQUID ENTHALPY AT CONCENTRATION X1.
      CALL HFIND(X1,HLQ,HLIQ,NENTH,XH)                                      0015
C     SLOPE AND Y-AXIS INTERCEPT OF INTERNAL REFLUX RATIO LINE.
      SLOPE=(DELD-HLQ)/(XD-X1)                                              0016
      CFACT=HLQ-SLOPE*X1                                                    0017
      DELY=0.01                                                             0018
      YG=X1                                                                 0019
      YG=YG+DELY                                                            0020
      NLOOP=130                                                             0021
      NCERR=0                                                               0022
```

```
C     'GO TO' LOOP 130 FINDS INTERSECTION OF VAPOUR ENTHALPY CURVE AND
C     INTERNAL REFLUX RATIO LINE,WHICH YIELDS THE Y-COORDINATE YG OF THE
C     OPERATING LINE POINT AT X1.
  130 CONTINUE                                                          0023
      NCERR=NCERR+1                                                     0024
      IF(NCERR.GT.500) GO TO 902                                        0025
      CALL HFIND(YG,HVP,HVAP,NENTH,YH)                                  0026
      HG=SLOPE*YG+CFACT                                                 0027
      IF(ABS(HVP-HG).LT.ACC) GO TO 110                                  0028
      NCALL=2                                                           0029
      CALL SEARCH(HVP,HG,YG,DELY,MN,NCALL)                              0030
      GO TO 130                                                         0031
  110 CONTINUE                                                          0032
      XROP(NPTS)=X1                                                     0033
      YROP(NPTS)=YG                                                     0034
      GO TO 100                                                         0035
  500 CONTINUE                                                          0036
C     STRIPPING SECTION.
C     END POINTS OF STRIPPING SECTION ARE XB AND XCR.
C     NEXT FIVE LINES CALCULATE THE SIZE AND NUMBER OF INCREMENTS IN
C     THE STRIPPING SECTION,KEEPING THE INCREMENT SIZE AS SIMILAR AS
C     POSSIBLE TO THAT USED IN THE RECTIFYING SECTION.
      DELBC=XCR-XB                                                      0037
      FACT=DELBC/H                                                      0038
      NFACT=IFIX(FACT)                                                  0039
      IF(NFACT.GT.NPPSC) GO TO 901                                      0040
      H=DELBC/NFACT                                                     0041
      MN=2                                                              0042
      X1=XB                                                             0043
      Y1=X1                                                             0044
      XSOP(1)=X1                                                        0045
      YSOP(1)=Y1                                                        0046
      MPTS=1                                                            0047
```

```
C     'GO TO' LOOP 200 USES PONCHON-SAVARIT METHOD TO FIND THE Y-COORDINATE
C     AT EACH VALUE OF X1,FROM XB TO XCR IN STEPS OF H.
  200 CONTINUE                                                          0048
      X1=X1+H                                                           0049
      IF(X1.GT.XCR) GO TO 999                                           0050
      MPTS=MPTS+1                                                       0051
C     'HFIND' RETURNS LIQUID ENTHALPY AT CONCENTRATION X1.
      CALL HFIND(X1,HLQ,HLIQ,NENTH,XH)                                  0052
C     SLOPE AND Y-AXIS INTERCEPT OF INTERNAL REFLUX RATIO LINE.
      SLOPE=(HLIQ-DELB)/(X1-XB)                                         0053
      CFACT=HLIQ-SLOPE*X1                                               0054
      DELY=0.01                                                         0055
      YG=Y1                                                             0056
      YG=YG+DELY                                                        0057
      NLOOP=230                                                         0058
      NCERR=0                                                           0059
C     'GO TO' LOOP 230 FINDS INTERSECTION OF VAPOUR ENTHALPY CURVE AND
C     INTERNAL REFLUX RATIO LINE,WHICH YIELDS THE Y-COORDINATE YG OF THE
C     OPERATING LINE POINT AT X1.
  230 CONTINUE                                                          0060
      NCERR=NCERR+1                                                     0061
      IF(NCERR.GT.500) GO TO 902                                        0062
      CALL HFIND(YG,HVP,HVAP,NENTH,YH)                                  0063
      HG=SLOPE*YG+CFACT                                                 0064
      IF(ABS(HVP-HG).LT.ACC) GO TO 210                                  0065
      NCALL=2                                                           0066
      CALL SEARCH(HVP,HG,YG,DELY,MN,NCALL)                              0067
      GO TO 230                                                         0068
  210 CONTINUE                                                          0069
      XSOP(MPTS)=X1                                                     0070
      YSOP(MPTS)=YG                                                     0071
      GO TO 200                                                         0072
```

```
C     ERROR MESSAGES ***************************************************
  901 CONTINUE                                                          0073
      WRITE(NTP,9001) NFACT                                             0074
 9001 FORMAT(/T2,'ERROR...NO. OF INTERVALS CALCULATED IN SUBROUTINE OPLI 0075
     1NE',/T2,'EXCEEDS DIMENSION SPECIFICATIONS: CALCULATED NO.=',I4)    0076
      MALF=1000                                                         0077
      GO TO 999                                                         0078
  902 CONTINUE                                                          0079
      WRITE(NTP,9002) NLOOP,HVP,HG,DELY                                 0080
 9002 FORMAT(/T2,'ERROR IN OPLINE,',/T2,'LOOP',I4,' EXCEEDED 500 ITERATI 0081
     1ONS.',/T2,'HVP=',E12.8,2X,'HG=',E12.8,2X,'Y-INCREMENT=',F10.8)     0082
      MALF=1000                                                         0083
  999 CONTINUE                                                          0084
      YSOP(MPTS)=YCR                                                    0085
      NPTS=NINT+1                                                       0086
      RETURN                                                            0087
      END                                                               0088

                                                        SUBROUTINE OPLINE
```

```
      SUBROUTINE FLOWS(NPTS,MPTS,NFIN,MFIN)                              0001
C     ROUTINE CALCULATES VAPOUR AND LIQUID FLOWRATES THROUGH A BINARY
C     DISTILLATION COLUMN WHEN CONSTANT MOLAL OVERFLOW IS NOT ASSUMED.
C     (PONCHON-SAVARIT DESIGN PROCEDURE)
C     ***********************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     ***********************************
      REAL LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                         0002
      NTOT=NPTS+MPTS                                                     0003
      IF(NTOT.GT.NPPSC) GO TO 901                                        0004
      NTOT1=NTOT-1                                                       0005
      NRFIN=MPTS+1                                                       0006
      YD=XD                                                              0007
      T1=TG                                                              0008
      LIQR(NTOT)=DTOT*R                                                  0009
      VAPR(NTOT)=DTOT*(R+1.)                                             0010
      CALL AVEMW(XD,YD,AVEL,AVEV,MW)                                     0011
      LRWT(NTOT)=LIQR(NTOT)*AVEL                                         0012
      VRWT(NTOT)=VAPR(NTOT)*AVEV                                         0013
      CALL BINBPT(A,B,C,NBDC,NDATA,PT,P1,TALF,TBP,T1,XALF,XD)            0014
      TRL(NTOT)=T1                                                       0015
      CALL BINDPT(A,B,C,NBDC,NDATA,PT,P1,TALF,TBP,T1,YALF,YD)            0016
      TRV(NTOT)=T1                                                       0017
```

```
C     'DO' LOOP 11 COMPUTES VAPOUR AND LIQUID MOLAR AND MASS FLOWRATES
C     IN RECTIFYING SECTION,TOGETHER WITH DEW AND BUBBLE POINTS.
      POSN='ABOVE'                                                        0018
      DO 11 I=NTOT1,NRFIN,-1                                              0019
      NRS=I-MPTS                                                          0020
      LIQR(I)=DTOT*(XD-YROP(NRS))/(YROP(NRS)-XROP(NRS))                   0021
      IF(LIQR(I).LT.0.0) GO TO 902                                        0022
      XAVE=XROP(NRS)                                                      0023
      YAVE=YROP(NRS)                                                      0024
      CALL AVEMW(XAVE,YAVE,AVEL,AVEV,MW)                                  0025
      LRWT(I)=LIQR(I)*AVEL                                                0026
      VAPR(I)=LIQR(I)*(XD-XROP(NRS))/(XD-YROP(NRS))                       0027
      IF(VAPR(I).LT.0.0) GO TO 903                                        0028
      VRWT(I)=VAPR(I)*AVEV                                                0029
      CALL BINBPT(A,B,C,NBDC,NDATA,PT,PI,TALF,TBP,TI,XALF,XAVE)           0030
      TRL(I)=TI                                                           0031
      CALL BINDPT(A,B,C,NBDC,NDATA,PT,PI,TALF,TBP,TI,YALF,YAVE)           0032
      TRV(I)=TI                                                           0033
   11 CONTINUE                                                            0034
C     'DO' LOOP 12 COMPUTES VAPOUR AND LIQUID MOLAR AND MASS FLOWRATES
C     IN STRIPPING SECTION,TOGETHER WITH DEW AND BUBBLE POINTS.
      POSN='BELOW'                                                        0035
      DO 12 I=MPTS,1,-1                                                   0036
      LIQS(I)=BTOT*(YSOP(I)-XB)/(YSOP(I)-XSOP(I))                         0037
      IF(LIQS(I).LT.0.0) GO TO 902                                        0038
      XAVE=XSOP(I)                                                        0039
      YAVE=YSOP(I)                                                        0040
      CALL AVEMW(XAVE,YAVE,AVEL,AVEV,MW)                                  0041
      LSWT(I)=LIQS(I)*AVEL                                                0042
      VAPS(I)=LIQS(I)*(XSOP(I)-XB)/(YSOP(I)-XB)                           0043
      IF(VAPS(I).LT.0.0) GO TO 903                                        0044
      VSWT(I)=VAPS(I)*AVEV                                                0045
      CALL BINBPT(A,B,C,NBDC,NDATA,PT,PI,TALF,TBP,TI,XALF,XAVE)           0046
      TSL(I)=TI                                                           0047
      CALL BINDPT(A,B,C,NBDC,NDATA,PT,PI,TALF,TBP,TI,YALF,YAVE)           0048
      TSV(I)=TI                                                           0049
   12 CONTINUE                                                            0050
```

```
C     REBOILER STAGE IS STEPPED OFF FROM XB AS IN A MCCABE-THIELE
C     CONSTRUCTION. STARTING POINT FOR THE PACKING IS XN,YN.
      CALL XYALFA(ALFA,ALFCON,ALF,MALF,NDATA,XALF,XB)                    0051
      YN=XB*ALF/(1.+XB*(ALF-1.))                                        0052
      IF(YN.LT.YSOP(MPTS)) GO TO 202                                    0053
      WRITE(NTP,2001) YN,XSOP(MPTS),YSOP(MPTS)                          0054
 2001 FORMAT(/T2,'THE COLUMN CONSISTS OF A RECTIFYING SECTION ONLY.',   0055
     1/T2,'COMPOSITION OF VAPOUR LEAVING REBOILER=',F7.4,/T2,'OPERATING 0056
     2LINES INTERSECT AT POINT (X,Y)=',F6.4,',',F6.4)                   0057
      IF(NOP.NE.2) GO TO 999                                            0058
      MFIN=1000                                                         0059
      DO 21 I=1,NPTS                                                    0060
      DIFF=YN-YROP(I)                                                   0061
      IF(DIFF.LT.0.0) GO TO 201                                         0062
   21 CONTINUE                                                          0063
  201 NFIN=I-1                                                          0064
      GO TO 301                                                         0065
  202 CONTINUE                                                          0066
      IF(NOP.NE.2) GO TO 999                                            0067
      DO 22 I=1,NPTS                                                    0068
      DIFF=YN-YSOP(I)                                                   0069
      IF(DIFF.LT.0.0) GO TO 203                                         0070
   22 CONTINUE                                                          0071
  203 MFIN=I-1                                                          0072
      NFIN=1                                                            0073
      GO TO 302                                                         0074
```

```
C     FLOWRATES AT THE BOTTOM OF THE PACKING.
  301 CONTINUE                                                          0075
      YFACT=(YROP(NFIN+1)-YN)/(YROP(NFIN+1)-YROP(NFIN))                 0076
      XN=XROP(NFIN+1)+(XROP(NFIN)-XROP(NFIN+1))*YFACT                   0077
      LIQR(NFIN)=BTOT*(YN-XB)/(YN-XN)                                   0078
      VAPR(NFIN)=LIQR(NFIN)*(XN-XB)/(YN-XB)                             0079
      CALL AVEMW(XN,YN,AVEL,AVEV,MW)                                    0080
      LRWT(NFIN)=LIQR(NFIN)*AVEL                                        0081
      VRWT(NFIN)=VAPR(NFIN)*AVEV                                        0082
      CALL BINBPT(A,B,C,NBDC,NDATA,PT,P1,TALF,TBP,TI,XALF,XN)           0083
      TRL(NFIN)=TI                                                      0084
      CALL BINDPT(A,B,C,NBDC,NDATA,PT,P1,TALF,TBP,TI,YALF,YN)           0085
      TRV(NFIN)=TI                                                      0086
      XROP(NFIN)=XN                                                     0087
      YROP(NFIN)=YN                                                     0088
      GO TO 999                                                         0089
  302 CONTINUE                                                          0090
      YFACT=(YSOP(MFIN+1)-YN)/(YSOP(MFIN+1)-YSOP(MFIN))                 0091
      XN=XSOP(MFIN+1)+(XSOP(MFIN)-XSOP(MFIN+1))*YFACT                   0092
      LIQS(MFIN)=BTOT*(YN-XB)/(YN-XN)                                   0093
      VAPS(MFIN)=LIQS(MFIN)*(XN-XB)/(YN-XB)                             0094
      CALL AVEMW(XN,YN,AVEL,AVEV,MW)                                    0095
      LSWT(MFIN)=LIQS(MFIN)*AVEL                                        0096
      VSWT(MFIN)=VAPS(MFIN)*AVEV                                        0097
      CALL BINBPT(A,B,C,NBDC,NDATA,PT,P1,TALF,TBP,TI,XALF,XN)           0098
      TSL(MFIN)=TI                                                      0099
      CALL BINDPT(A,B,C,NBDC,NDATA,PT,P1,TALF,TBP,TI,YALF,YN)           0100
      TSV(MFIN)=TI                                                      0101
      XSOP(MFIN)=XN                                                     0102
      YSOP(MFIN)=YN                                                     0103
      GO TO 999                                                         0104
```

```
C     ERROR MESSAGES ***************************************************************
  901 CONTINUE                                                                    0105
      WRITE(NTP,9001) NTOT                                                        0106
 9001 FORMAT(/T2,'TOTAL NUMBER OF OPERATING LINE DATA POINTS.',I5,                0107
     1/T2,'EXCEEDS DIMENSION SPECIFICATIONS.')                                    0108
      MALF=1000                                                                   0109
      GO TO 999                                                                   0110
  902 CONTINUE                                                                    0111
      WRITE(NTP,9002) POSN,I                                                      0112
 9002 FORMAT(/T2,'NEGATIVE LIQUID FLOW DETECTED ',A5,' FEED POSITION.',          0113
     1/T2,'CHANGE IN FEED Q VALUE OR REFLUX RATIO SUGGESTED.'//T2,               0114
     2'POINT OF OCCURENCE ON OPERATING LINE:',I4)                                0115
      MALF=1000                                                                   0116
      GO TO 999                                                                   0117
  903 CONTINUE                                                                    0118
      WRITE(NTP,9003) POSN,I                                                      0119
 9003 FORMAT(/T2,'NEGATIVE VAPOUR FLOW DETECTED ',A5,' FEED POSITION.',          0120
     1/T2,'CHANGE IN FEED Q VALUE OR REFLUX RATIO SUGGESTED.'//T2,               0121
     2'POINT OF OCCURENCE ON OPERATING LINE:',I4)                                0122
      MALF=1000                                                                   0123
  999 CONTINUE                                                                    0124
      RETURN                                                                      0125
      END                                                                         0126
```

```
      SUBROUTINE AVEMW(X1,Y1,AVEL,AVEV,MW)                              0001
      REAL MW                                                           0002
      DIMENSION MW(2)                                                   0003
      X2=1.-X1                                                          0004
      Y2=1.-Y1                                                          0005
      AVEL=X1*MW(1)+X2*MW(2)                                            0006
      AVEV=Y1*MW(1)+Y2*MW(2)                                            0007
      RETURN                                                            0008
      END                                                               0009
```

```
      SUBROUTINE CHEC(ALFA,ALFCON,FLOW,MALF,NDATA,NN,NTP,XALF,XQ,YQ,ZFS)     0001
C     ROUTINE CHECKS THAT ALL OPERATING LINE INTERSECTION POINTS LIE
C     BETWEEN THE EQUILIBRIUM CURVE AND THE DIAGONAL ON THE X-Y DIAGRAM.
C     THIS ENSURES THAT NO PINCH POINTS EXIST.
      DIMENSION ALFA(1),FLOW(1),XALF(1),XQ(1),YQ(1),ZFS(1)                  0002
      DO 10 I=2,NN                                                          0003
      XA=XQ(I)                                                              0004
      CALL XYALFA(ALFA,ALFCON,ALF,MALF,NDATA,XALF,XA)                       0005
      YA=XA*ALF/(1.+XA*(ALF-1.))                                            0006
      IF(YA.GT.YQ(I)) GO TO 10                                              0007
      IF(FLOW(I)) 101,102,102                                               0008
  101 CONTINUE                                                              0009
      WRITE(NTP,1011) ZFS(I)                                                0010
 1011 FORMAT(/T2,'PINCH POINT AT SIDE STREAM:',/T2,'COMPOSITION',F7.4)      0011
      GO TO 110                                                             0012
  102 CONTINUE                                                              0013
      WRITE(NTP,1021) ZFS(I)                                                0014
 1021 FORMAT(/T2,'PINCH POINT AT FEED STREAM:',/T2,'COMPOSITION',F7.4)      0015
      GO TO 110                                                             0016
   10 CONTINUE                                                              0017
      GO TO 999                                                             0018
  110 CONTINUE                                                              0019
      MALF=1000                                                             0020
      NN1=NN+1                                                              0021
  999 CONTINUE                                                              0022
      RETURN                                                                0023
      END                                                                   0024
```

SUBROUTINE CHEC

```
      SUBROUTINE STEPD(NTC,NPTS,MPTS)                                    0001
C     ROUTINE FOR STEPPING OFF STAGES IN MCCABE-THIELE OR PONCHON-SAVARIT
C     DESIGN OF BINARY DISTILLATION TRAY-COLUMNS.
C     ROUTINE COMPUTES: (1) TOTAL NUMBER OF THEORETICAL TRAYS REQUIRED.
C                       (2) TRAY POSITIONS OF FEED AND SIDE STREAMS.
C                       (3) FLOW AND TEMP. PROFILES FOR P-S DESIGNS.
C     **********************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     **********************************
      REAL LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                         0002
      DATA I,MT,NI,NP/1,0,2,0/                                           0003
      NN1=NN+1                                                           0004
      X(1)=XB                                                            0005
      Y(1)=YB                                                            0006
      XA1=XB                                                             0007
      YA1=YB                                                             0008
      XA=XB                                                              0009
C     'DO' LOOP 10 CAUSES STEPPING PROCEDURE TO MOVE FROM ONE OPERATING
C     LINE TO THE NEXT SO AS TO MINIMISE THE TOTAL NUMBER OF THEORETICAL
C     STAGES.
C     FEED AND SIDE STREAMS ARE ALSO POSITIONED HERE.
      DO 10 J=2,NN1                                                      0010
      IF(Y(I).GT.QY(J)) GO TO 100                                        0011
C     'DO' LOOP 11 COMPUTES AND STORES THE COORDINATES OF EACH POINT
C     IN THE STEPPING PROCEDURE AND COUNTS THE NUMBER OF STAGES (MT).
      DO 11 I=NI,100,2                                                   0012
      MT=MT+1                                                            0013
      IF(I.GT.NPTCC) GO TO 901                                           0014
      X(I)=X(I-1)                                                        0015
      XA=X(I)                                                            0016
      CALL XYALFA(ALFA,ALFCON,ALPHA,MALF,NDATA,XALF,XA)                  0017
      Y(I)=(XA*ALPHA)/(1.+XA*(ALPHA-1.))                                 0018
      IF(MT.GT.1) Y(I)=Y(I-1)+(Y(I)-Y(I-1))*EFF                          0019
      IF(Y(I).GT.QY(J)) GO TO 100                                        0020
      IF(Y(I).GT.YC) GO TO 200                                           0021
```

```
C       X(I+1) FOR MCCABE-THIELE DESIGNS.
        IF(NOP.EQ.1) X(I+1)=(Y(I)-CFCT(J-1))/SLPE(J-1)                    0022
        IF(NOP.EQ.1) GO TO 111                                            0023
C       X(I+1) FOR PONCHON-SAVARIT DESIGNS.
        CALL PSOPXY(Y(I),XOUT,XB,XCR,XROP,XSOP,YROP,YSOP,YB,YC,YCR)       0024
        X(I+1)=XOUT                                                       0025
  111   CONTINUE                                                          0026
        Y(I+1)=Y(I)                                                       0027
   11   CONTINUE                                                          0028
  100   CONTINUE                                                          0029
        IF(NOP.GT.1) GO TO 101                                            0030
C       X(I+1) FOR MCCABE-THIELE DESIGNS.
        X(I+1)=(Y(I)-CFCT(J))/SLPE(J)                                     0031
        GO TO 102                                                         0032
  101   CONTINUE                                                          0033
C       X(I+1) FOR PONCHON-SAVARIT DESIGNS.
        CALL PSOPXY(Y(I),XOUT,XB,XCR,XROP,XSOP,YROP,YSOP,YB,YC,YCR)       0034
        X(I+1)=XOUT                                                       0035
  102   CONTINUE                                                          0036
        Y(I+1)=Y(I)                                                       0037
        NP=NP+1                                                           0038
C       TRAY NUMBER POSITIONS OF FEED AND SIDE STREAMS.
        NTRAY(NP)=(I/2)-1                                                 0039
        IF(Y(I).GT.YC) GO TO 200                                          0040
        NI=I+2                                                            0041
   10   CONTINUE                                                          0042
  200   CONTINUE                                                          0043
        TFRAC=(YC-Y(I-2))/(Y(I)-Y(I-2))                                   0044
        TRAYNO=(NT-(1+NCOND))+TFRAC                                       0045
        NTC=I+1                                                           0046
        IF(NOP.NE.3) GO TO 999                                            0047
```

```
C     TEMPERATURE AND VAPOUR-LIQUID FLOWRATE PROFILES FOR P-S DESIGNS.
      DO 20 I=1,NTC                                                        0048
      XA=X(I)                                                              0049
      IF(XA.GT.XCR) GO TO 201                                              0050
C     STRIPPING SECTION.
      NP=MPTS                                                              0051
      MP=0                                                                 0052
      CALL FFIND(XA,FLOV,FLOL,TEMP,VAPS,LIQS,TSL,XSOP,NP,MP)               0053
      LRWT(I)=FLOL                                                         0054
      VRWT(I)=FLOV                                                         0055
      TRV(I)=TEMP                                                          0056
      GO TO 20                                                             0057
  201 CONTINUE                                                             0058
C     RECTIFYING SECTION.
      NP=NPTS                                                              0059
      MP=MPTS                                                              0060
      CALL FFIND(XA,FLOV,FLOL,TEMP,VAPR,LIQR,TRL,XROP,NP,MP)               0061
      LRWT(I)=FLOL                                                         0062
      VRWT(I)=FLOV                                                         0063
      TRV(I)=TEMP                                                          0064
   20 CONTINUE                                                             0065
      LRWT(1)=BTOT                                                         0066
      IF(NCOND.EQ.1) VRWT(NTC)=0.0                                         0067
      IF(NCOND.EQ.2) VRWT(NTC)=DTOT                                        0068
      GO TO 999                                                            0069
C     ERROR MESSAGES ***************************************************
  901 CONTINUE                                                             0070
      WRITE(NTP,9001) MT                                                   0071
 9001 FORMAT(/T2,'NUMBER OF TRAYS EXCEEDS:',I5,/T2,'PINCH ZONE IS SUSPEC   0072
     1TED.'/T2,'CURRENT RESULTS WILL BE PRINTED AND THESE SHOULD BE'       0073
     2/T2,'USED TO CONSTRUCT A MCCABE-THIELE DIAGRAM.',/T2,'IF A PINCH Z   0074
     3ONE IS NOT OBVIOUS THE SIZE OF THE',/T2,'ARRAYS X AND Y IN THE COM   0075
     4MON BLOCK MUST BE INCREASED TO',/T2,'ALLOW FOR A GREATER NUMBER OF   0076
     5 TRAYS.')                                                            0077
  999 CONTINUE                                                             0078
      RETURN                                                               0079
      END                                                                  0080
```

```
      SUBROUTINE PSOPXY(Y1,XOUT,XB,XCR,XROP,XSOP,YROP,YSOP,YB,YC,YCR)        0001
C     ROUTINE SEARCHES FOR X-Y COORDINATES ON PONCHON-SAVARIT OPERATING
C     LINES FOR STEPPING OFF THEORETICAL STAGES IN 'STEPD'.
C     LINEAR INTERPOLATION IS USED BETWEEN ACTUAL DATA POINTS.
      DIMENSION XROP(1),XSOP(1),YROP(1),YSOP(1)                             0002
      IF(Y1.GT.YSOP(2)) GO TO 100                                           0003
      YFACT=(Y1-YB)/(YSOP(2)-YB)                                            0004
      XOUT=XB+(XSOP(2)-XB)*YFACT                                            0005
      GO TO 999                                                             0006
  100 CONTINUE                                                              0007
      IF(Y1.GT.YCR) GO TO 200                                               0008
      DO 10 I=2,500                                                         0009
      DIFF=Y1-YSOP(I)                                                       0010
      IF(DIFF.LE.0.0) GO TO 110                                             0011
   10 CONTINUE                                                              0012
  110 YFACT=(YSOP(I)-Y1)/(YSOP(I)-YSOP(I-1))                                0013
      XOUT=XSOP(I)+(XSOP(I-1)-XSOP(I))*YFACT                                0014
      GO TO 999                                                             0015
  200 CONTINUE                                                              0016
      IF(Y1.GT.YROP(2)) GO TO 201                                           0017
      YFACT=(Y1-YCR)/(YROP(2)-YCR)                                          0018
      XOUT=XCR+(XROP(2)-XCR)*YFACT                                          0019
      GO TO 999                                                             0020
  201 CONTINUE                                                              0021
      IF(Y1.GT.YC) GO TO 300                                                0022
      DO 20 I=2,500                                                         0023
      DIFF=Y1-YROP(I)                                                       0024
      IF(DIFF.LE.0.0) GO TO 220                                             0025
   20 CONTINUE                                                              0026
  220 YFACT=(YROP(I)-Y1)/(YROP(I)-YROP(I-1))                                0027
      XOUT=XROP(I)+(XROP(I-1)-XROP(I))*YFACT                                0028
      GO TO 999                                                             0029
  300 CONTINUE                                                              0030
      XOUT=Y1                                                               0031
  999 CONTINUE                                                              0032
      RETURN                                                                0033
      END                                                                   0034
```

```
      SUBROUTINE FFIND(XA,FLOV,FLOL,TEMP,FVAP,FLIQ,FTEMP,FX,NPTS,MPTS)      0001
C     ROUTINE CALCULATES LIQUID AND VAPOUR FLOWS RESPECTIVELY LEAVING
C     AND ENTERING THEORETICAL STAGES IN PONCHON-SAVARIT DESIGNS.
C     LINEAR INTERPOLATION IS USED BETWEEN ACTUAL DATA POINTS.
      DIMENSION FVAP(1),FLIQ(1),FTEMP(1),FX(1)                              0002
      NTOT=NPTS+MPTS                                                        0003
      MPTS1=MPTS+2                                                          0004
      IF(XA.GT.FX(2)) GO TO 100                                             0005
      XFACT=(FX(2)-XA)/(FX(2)-FX(1))                                        0006
      FLOV=FVAP(MPTS1)-(FVAP(MPTS1+1)-FVAP(MPTS1))*XFACT                    0007
      FLOL=FLIQ(MPTS1)-(FLIQ(MPTS1+1)-FLIQ(MPTS1))*XFACT                    0008
      TEMP=FTEMP(MPTS1)-(FTEMP(MPTS1+1)-FTEMP(MPTS1))*XFACT                 0009
      GO TO 999                                                             0010
  100 CONTINUE                                                              0011
      DO 10 I=MPTS1,NTOT                                                    0012
      NNO=I-MPTS                                                            0013
      DIFF=XA-FX(NNO)                                                       0014
      IF(DIFF.LE.0.0) GO TO 110                                             0015
   10 CONTINUE                                                              0016
      GO TO 120                                                             0017
  110 CONTINUE                                                              0018
      XFACT=(FX(NNO)-XA)/(FX(NNO)-FX(NNO-1))                                0019
      FLOV=FVAP(I)+(FVAP(I-1)-FVAP(I))*XFACT                                0020
      FLOL=FLIQ(I)+(FLIQ(I-1)-FLIQ(I))*XFACT                                0021
      TEMP=FTEMP(I)+(FTEMP(I-1)-FTEMP(I))*XFACT                             0022
      GO TO 999                                                             0023
  120 CONTINUE                                                              0024
      FLOV=FVAP(NTOT)                                                       0025
      FLOL=FLIQ(NTOT)                                                       0026
      TEMP=FTEMP(NTOT)                                                      0027
  999 CONTINUE                                                              0028
      RETURN                                                                0029
      END                                                                   0030

                                                        SUBROUTINE FFIND
```

```
      SUBROUTINE BINRES(NDAT,NTC,NN1,NPTS,MPTS)                          0001
C     ROUTINE FOR PRINTING RESULTS OF MCCABE-THIELE OR PONCHON-SAVARIT
C     SOLUTIONS TO BINARY CONTINUOUS-DISTILLATION PROBLEMS.
C     ************************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     ************************************
      REAL LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                         0002
      CON='COND'                                                         0003
    1 FORMAT(A3)                                                         0004
 1001 FORMAT(/T2,'MCCABE-THIELE DESIGN OF A BINARY DISTILLATION SYSTEM,'  0005
     1)                                                                  0006
 1011 FORMAT(/T2,'PONCHON-SAVARIT DESIGN OF A BINARY DISTILLATION SYSTEM  0007
     1.')                                                                0008
 1002 FORMAT(/T2,'FEED COMPONENTS: (1) ',5A6,/T19,'(2) ',5A6)            0009
 1003 FORMAT(/T2,'NO. OF FEED STREAMS:',I4,/T2,'NO. OF SIDE STREAMS:',   0010
     1I4)                                                                0011
 1004 FORMAT(/T2,'NO. OF THEORETICAL STAGES REQUIRED:',F6.1)             0012
 1005 FORMAT(/T2,'MURPHREE EFFICIENCY FACTOR USED:',F5.2)                0013
 1006 FORMAT(/T2,'FEED AND SIDE STREAM SPECIFICATIONS:',/T2,             0014
     1'STREAM',2X,'TRAY NO',2X,'Q VALUE',2X,'FLOWRATE',2X,'COMPOSITION', 0015
     2/T2,'NO',1X,'TYPE',T28,'KGMOLE/HR',3X,'MOL FRAC')                  0016
 1007 FORMAT(/T2,'DISTILLATE WITHDRAWAL RATE:',F7.2,' KGMOLE/HR',/T2,    0017
     1'DISTILLATE COMPOSITION:',6X,F7.4)                                 0018
 1008 FORMAT(/T2,'BOTTOMS WITHDRAWAL RATE:',1X,F7.2,' KGMOLE/HR',/T2,    0019
     1'BOTTOMS COMPOSITION:',7X,F7.4)                                    0020
 1009 FORMAT(T2,'OPEN STEAM OPERATION.',/T2,'SATURATED STEAM REQUIREMENT  0021
     1S:',F7.2,' KGMOLE/HR')                                             0022
 1110 FORMAT(T2,I2,1X,'SIDE',3X,I3,5X,F5.2,4X,F7.2,5X,F6.4)              0023
 1120 FORMAT(T2,I2,1X,'FEED',3X,I3,5X,F5.2,4X,F7.2,5X,F6.4)              0024
 1201 FORMAT(/T2,'TOTAL CONDENSER OPERATION.')                           0025
 1202 FORMAT(/T2,'PARTIAL CONDENSER OPERATION.')                         0026
 1203 FORMAT(T2,'CONDENSER PRESSURE:',F7.2,1X,'KPA')                     0027
 1204 FORMAT(T2,'FEED TEMPERATURE:',F7.2,1X,'DEG C',/T2,'FEED ENTHALPY', 0028
     1F11.2,1X,'KJ/KGMOLE')                                              0029
 1205 FORMAT(T2,'CONDENSER DUTY:',F11.2,1X,'KW')                         0030
 1206 FORMAT(T2,'REBOILER DUTY:',8X,F10.2,1X,'KW')                       0031
 1207 FORMAT(/T2,'REFLUX RATIO:',F11.4)                                  0032
 1208 FORMAT(T2,'MIN REFLUX RATIO:',F7.4)                                0033
```

```
 1310 FORMAT(/T2,'VAPOUR AND LIQUID RATES THROUGH COLUMN:',/T35,          0034
     1'VAPOUR',3X,'LIQUID',3X,'(KGMOLE/HR)')                             0035
 1311 FORMAT(T2,'SECTION',I2,'  REBOILER - ',A4,' (',I1,')',3X,F6.2,3X,  0036
     1F6.2)                                                              0037
 1312 FORMAT(T2,'SECTION',I2,2X,A4,' (',I1,') - ',A4,' (',I1,')',3X,F6.2 0038
     1,3X,F6.2)                                                          0039
 1313 FORMAT(T2,'SECTION',I2,2X,A4,' (',I1,') - CONDENSER',2X,F6.2,3X,   0040
     1F6.2)                                                              0041
 1320 FORMAT(/T2,'COMPOSITION,TEMPERATURE AND FLOWRATE PROFILES:')       0042
 1321 FORMAT(T2,'TRAY',3X,'COMPOSITION',5X,'TEMPERATURE',6X,'FLOWRATES'  0043
     1,/T9,'(MOLE FRAC)',7X,'(DEG C)',7X,'(KGMOLE/HR)',/T9,              0044
     2'LIQ',5X,'VAP',6X,'BUBBLE PT',6X,'LIQ',5X,'VAP')                   0045
 1322 FORMAT(T2,I3,2X,F6.4,2X,F6.4,6X,F7.2,4X,F7.2,1X,F7.2)              0046
 1323 FORMAT(T2,'FEED TRAY POSITION:')                                   0047
 1324 FORMAT(T2,A4,1X,F6.4,2X,F6.4,6X,F7.2,4X,F7.2,1X,F7.2)              0048
 1400 FORMAT(/T2,'DO YOU WANT OPERATING DIAGRAM DATA TABLES?')           0049
 1401 FORMAT(T5,F6.4,2X,F6.4)                                            0050
 1410 FORMAT(/T2,'EQUILIBRIUM DATA:',/T8,'X',7X,'Y')                     0051
 1411 FORMAT(/T2,'ENTHALPY DIAGRAM DEL PTS:',/T4,'DEL D=',E12.6,         0052
     1X,'KJ/KGMOLE',/T4,'DEL B=',E12.6,1X,'KJ/KGMOLE')                   0053
 1420 FORMAT(/T2,'INTERSECTION COORDINATES OF OPERATING LINES:',/T8,     0054
     1'X',7X,'Y')                                                        0055
 1421 FORMAT(/T2,'POINTS ON OPERATING LINES:',/T8,'X',7X,'Y')            0056
 1430 FORMAT(/T2,'COORDINATES OF MCCABE-THIELE STAGES:',/T8,'X',7X,'Y')  0057
C      RESULT TABLE HEADING,COMPONENT NAMES AND GENERAL DATA.
C      MCCABE-THIELE DESIGNS.
       IF(NOP.EQ.1) WRITE(NTP,1001)                                      0058
C      PONCHON-SAVARIT DESIGNS.
       IF(NOP.EQ.3) WRITE(NTP,1011)                                      0059
       WRITE(NTP,1002) ((TITLES(I,J),J=1,5),I=1,2)                       0060
       WRITE(NTP,1003) NF,NS                                             0061
       WRITE(NTP,1004) TRAYNO                                            0062
       WRITE(NTP,1005) EFF                                               0063
       WRITE(NTP,1006)                                                   0064
```

```
C     FEED AND SIDE STREAM SPECIFICATIONS,AND POSITIONS.
      DO 10 I=2,NN                                                    0065
      I1=I-1                                                          0066
      IF(FLOW(I)) 110,10,120                                          0067
  110 FLO=ABS(FLOW(I))                                                0068
      WRITE(NTP,1110) I1,NTRAY(I-1),QFS(I),FLO,ZFS(I)                 0069
      GO TO 10                                                        0070
  120 WRITE(NTP,1120) I1,NTRAY(I-1),QFS(I),FLOW(I),ZFS(I)             0071
   10 CONTINUE                                                        0072
C     FEED TEMPERATURE AND ENTHALPY FOR P-S DESIGNS.
      IF(NOP.EQ.3) WRITE(NTP,1204) TF,QF(1)                           0073
C      CONDENSER SPECIFICATIONS.
      IF(NCOND.EQ.1) WRITE(NTP,1201)                                  0074
      IF(NCOND.EQ.2) WRITE(NTP,1202)                                  0075
      PT1=PT/7.50061683                                               0076
      WRITE(NTP,1203) PT1                                             0077
      IF(NOP.EQ.3) WRITE(NTP,1205) QC                                 0078
C     REFLUX CONDITIONS.
      WRITE(NTP,1207) R                                               0079
      IF(NOP.EQ.3) WRITE(NTP,1208) RM                                 0080
C     DISTILLATE PRODUCT DATA.
      WRITE(NTP,1007) DTOT,XD                                         0081
C     BOTTOMS PRODUCT DATA,AND REBOILER SPECIFICATIONS.
      WRITE(NTP,1008) BTOT,XB                                         0082
      IF(NOS.EQ.1) WRITE(NTP,1009) AV(1)                              0083
      IF(NOP.EQ.3) WRITE(NTP,1206) QB                                 0084
      IF(NOP.EQ.3) GO TO 131                                          0085
C     FLOWRATES IN EACH SECTION OF COLUMN FOR M-T DESIGNS.
      WRITE(NTP,1310)                                                 0086
      NI=1                                                            0087
      IF(FLOW(2).LT.0.0) STR='SIDE'                                   0088
      IF(FLOW(2).GT.0.0) STR='FEED'                                   0089
      WRITE(NTP,1311) NI,STR,NI,AV(1),AL(1)                           0090
      NNI=NN-1                                                        0091
      IF(NNI.LT.2) GO TO 130                                          0092
```

```
      DO 11 I=2,NNI                                                    0093
      NI=NI+1                                                          0094
      NIM=NI-1                                                         0095
      STR1=STR                                                         0096
      IF(FLOW(I+1).LT.0.0) STR='SIDE'                                  0097
      IF(FLOW(I+1).GT.0.0) STR='FEED'                                  0098
      WRITE(NTP,1312) I,STR1,NIM,STR,NI,AV(I),AL(I)                    0099
   11 CONTINUE                                                         0100
  130 CONTINUE                                                         0101
      I=NI+1                                                           0102
      WRITE(NTP,1313) I,STR,NI,AV(NN),AL(NN)                           0103
      GO TO 132                                                        0104
  131 CONTINUE                                                         0105
C     COMPOSITION,TEMPERATURE AND FLOWRATE PROFILES FOR P-S DESIGNS.
      WRITE(NTP,1320)                                                  0106
      WRITE(NTP,1321)                                                  0107
      NTRA=(NTRAY(1)+1)*2                                              0108
      MI=-1                                                            0109
      DO 30 I=1,NTRA,2                                                 0110
      MI=MI+1                                                          0111
      WRITE(NTP,1322) MI,X(I),Y(I),TRV(I),LRWT(I),VRWT(I)              0112
   30 CONTINUE                                                         0113
      NTRA=NTRA+1                                                      0114
      WRITE(NTP,1323)                                                  0115
      DO 31 I=NTRA,NTC,2                                               0116
      MI=MI+1                                                          0117
      IF(I.EQ.NTC) GO TO 133                                           0118
      WRITE(NTP,1322) MI,X(I),Y(I),TRV(I),LRWT(I),VRWT(I)              0119
      GO TO 31                                                         0120
  133 WRITE(NTP,1324) CON,X(I),Y(I),TRV(I),LRWT(I),VRWT(I)             0121
   31 CONTINUE                                                         0122
```

```
  132 CONTINUE                                                          0123
C     ENTHALPY_CONCENTRATION DIAGRAM 'DEL' POINTS.
      IF(NOP.EQ.3) WRITE(NTP,1411) DELD,DELB                            0124
      WRITE(NTP,1400)                                                   0125
      READ(NCR,1) ANAME                                                 0126
      IF(ANAME.NE.'YES') GO TO 999                                      0127
C     EQUILIBRIUM CURVE DATA.
      WRITE(NTP,1410)                                                   0128
      WRITE(NTP,1401) (XEQUIL(I),YEQUIL(I),I=1,NDAT,5)                  0129
      IF(NOP.NE.3) GO TO 141                                            0130
C     OPERATING LINE DATA FOR PONCHON-SAVARIT DESIGNS.
      WRITE(NTP,1421)                                                   0131
      WRITE(NTP,1401) (XSOP(I),YSOP(I),I=1,MPTS,5)                      0132
      WRITE(NTP,1323)                                                   0133
      WRITE(NTP,1401) (XROP(I),YROP(I),I=1,NPTS,5)                      0134
  141 CONTINUE                                                          0135
C     OPERATING LINE POINTS OF INTERSECTION.
      WRITE(NTP,1420)                                                   0136
      WRITE(NTP,1401) (QX(I),QY(I),I=1,NN1)                             0137
C     COORDINATES FOR CONSTRUCTING STAGES ON X-Y DIAGRAM.
      WRITE(NTP,1430)                                                   0138
      WRITE(NTP,1401) (X(I),Y(I),I=1,NTC)                               0139
  999 CONTINUE                                                          0140
      RETURN                                                            0141
      END                                                               0142
```

```
C     COMMON BLOCK 'PACCOM' FOR PACKED-COLUMN DESIGN PROGRAM.
      PARAMETER NPEQ=55                                                    0001
      PARAMETER NPPS=150                                                   0002
      PARAMETER NPTC=55                                                    0003
      PARAMETER NPEN=25                                                    0004
      PARAMETER NPAK=225                                                   0005
      DIMENSION ASURF(NPAK),CFAK(NPAK),FUNCR(NPAK),FUNCS(NPAK)             0006
      DIMENSION KX(NPAK),KY(NPAK),L(NPAK),SLOPK(NPAK)                      0007
      DIMENSION V(NPAK),VYR(NPAK),VYS(NPAK),XI(NPAK),YI(NPAK)              0008
      DIMENSION ALFA(NPEQ),HLIQ(NPEN),HVAP(NPEN),LIQR(NPPS),LIQS(NPPS)     0009
      DIMENSION LRWT(NPPS),LSWT(NPPS),TALF(NPEQ)                           0010
      DIMENSION TRL(NPPS),TRV(NPPS),TSL(NPPS),TSV(NPPS)                    0011
      DIMENSION VAPR(NPPS),VAPS(NPPS),VRWT(NPPS),VSWT(NPPS)                0012
      DIMENSION X(NPTC),XALF(NPEQ),XEQUIL(NPEQ),XH(NPEN)                   0013
      DIMENSION XROP(NPPS),XSOP(NPPS),YROP(NPPS),YSOP(NPPS)                0014
      DIMENSION Y(NPTC),YALF(NPEQ),YEQUIL(NPEQ),YH(NPEN)                   0015
      COMMON A(2),ACP(2),AL(11),ALFA,ALFCON,AV(11)                         0016
      COMMON B(2),BCP(2),BTOT,C(2),CCP(2),CFCT(12),CPCONV,CPMW(2)          0017
      COMMON DCP(2),DELB,DELD,DL,DTOT,DV                                   0018
      COMMON EFF,FLOW(10),FTOT,FTOTS(5)                                    0019
      COMMON HB,HD,HEVABS,HF,HLIQ,HSCONV,HSMW(2),HSOLN(30)                 0020
      COMMON HVAP,LH(2)                                                    0021
      COMMON LHCONV,LHMW(2),LIQR,LIQS,LRWT,LSWT                            0022
      COMMON MALF,MW(2),NBDC,NCOND,NDATA,NENTH,NF,NHS,NINT                 0023
      COMMON NN,NOP,NOS,NPPSC,NPTCC,NS,NTRAY(12)                           0024
      COMMON NCR,NGI,NTP                                                   0025
      COMMON PT,Q,QB,QC,QF(5),QS(5),QFS(10),QX(12),QY(12),R,RM             0026
      COMMON SLPE(12),STOTS(5),TALF,TBASE,TBP(2),TC(2)                     0027
      COMMON TCONV,TF,TG,TITLES(2,5),TRAYNO,TRL,TRV                        0028
      COMMON TSL,TSV,VAPR,VAPS,VRWT,VSWT                                   0029
      COMMON X,XALF,XB,XC,XCR,XD,XEQUIL,XH,XHS(30)                         0030
      COMMON XROP,XSOP,YROP,YSOP                                           0031
      COMMON Y,YALF,YB,YC,YCR,YD,YEQUIL,YH                                 0032
      COMMON Z,ZF(5),ZFI(5),ZS(5),ZSI(5),ZFS(10)                           0033
```

```
C     MAINLINE PROGRAM FOR DESIGN OF PACKED COLUMNS FOR USE IN BINARY
C     CONTINUOUS-DISTILLATION OPERATIONS.
C     THE PACKAGE INCLUDES DATA FOR CERAMIC RASCHIG RINGS AND BERL SADDLES
C     (SOURCE: SHULMAN FROM TREYBAL''MASS TRANSFER OPERATIONS'',2ND ED)
C     SPECIFICATIONS FOR PACKINGS OTHER THAN THESE MUST BE SUPPLIED BY
C     THE USER.
C     **********************************
C     INSERT COMMON BLOCK 'PACCOM' HERE.
C     **********************************
      REAL KX,KY,L,LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                    0001
    1 FORMAT( )                                                             0002
    2 FORMAT(A3)                                                            0003
    3 FORMAT(5A6)                                                           0004
      DATA NCR,NGI,NTP/8,11,5/                                              0005
      DATA NCODE,NINT,NOP/3,50,2/                                           0006
      DATA MALF,MCAN/1,1/                                                   0007
      DATA ALBKGH,ASFSM,AMLCON,AMVCON/4,8824.0,3048,2,42,0.000242/          0008
      DATA DLCON,DVCON,RHOCON,SIGCON/3.87504,3.87504,62.43,1.0/             0009
      GC=4.18*(10.**8)                                                      0010
      NPPSC=NPPS-2                                                          0011
      NPTCC=NPTC                                                            0012
C     START OF PROGRAM ***************************************************
      WRITE(NTP,5000)                                                       0013
 5000 FORMAT(/T2,'ANSWER ALL QUESTIONS YES OR NO.'//T2,'ALL INPUT IN FRE    0014
     1E FORMAT.',//T2,'DO YOU KNOW HOW TO USE THIS PROGRAM?')               0015
      READ(NCR,2) ANAME                                                     0016
      IF(ANAME.EQ.'YES') GO TO 500                                          0017
      CALL TEACH(NCODE)                                                     0018
      GO TO 999                                                             0019
  500 CONTINUE                                                              0020
      WRITE(NTP,5001)                                                       0021
 5001 FORMAT(/T2,'DO YOU WANT TO ADD DATA ELEMENTS?')                       0022
      READ(NCR,2) CNAME                                                     0023
      IF(CNAME.EQ.'NO') GO TO 501                                           0024
```

```
      MCAN=2                                                                0025
      WRITE(NTP,5002)                                                       0026
 5002 FORMAT(/T2,'ADD YOUR FIRST DATA ELEMENT.')                            0027
  501 CONTINUE                                                              0028
C     'READBD' READS DATA ELEMENT.
      CALL READBD(MCAN,NDAT)                                                0029
      IF(MALF.GT.999) GO TO 999                                             0030
C     START OF COMPUTATION **************************************************
      WRITE(NTP,1001)                                                       0031
 1001 FORMAT(/T2,'DATA ACCEPTED. START OF OPERATING DIAGRAM CALCS.')         0032
C     'PONSAV' CONSTRUCTS THE PONCHON-SAVARIT OPERATING DIAGRAM.
      CALL PONSAV(NPTS,MPTS,NFIN,MFIN)                                      0033
      IF(MALF.GT.999) GO TO 999                                             0034
      NTOT=NPTS+MPTS                                                        0035
      MPTI=MPTS+NFIN                                                        0036
C
C     START OF PACKING SPECS ************************************************
C
      WRITE(NTP,2001)                                                       0037
 2001 FORMAT(/T2,'OPERATING DIAGRAM COMPLETE.',/T2,'START OF PACKING CAL    0038
     1CULATIONS.')                                                          0039
      WRITE(NTP,5030)                                                       0040
 5030 FORMAT(/T2,'DO YOU WANT TO ADD A SECOND DATA ELEMENT?')               0041
      READ(NCR,2) CNAME                                                     0042
      MCAN=2                                                                0043
      IF(CNAME.NE.'YES') MCAN=1                                             0044
      IF(MCAN.NE.1) WRITE(NTP,5031)                                         0045
 5031 FORMAT(/T2,'ADD YOUR SECOND DATA ELEMENT.')                           0046
      CALL PACTYP(MCAN,NTYPE,NPACK,NCR,NTP)                                 0047
      IF(MCAN.GT.99) GO TO 999                                              0048
C     'PAKING' RETURNS DESIGN DATA CORRESPONDING TO SPECIFIED TYPE
C     OF PACKING. (EITHER RASCHIG RINGS OR BERL SADDLES)
      CALL PAKING(NTYPE,NPACK,CF,E,AMF1,AMF2,ANF1A,ANF1B,ANF2A,ANF2B,PF1    0049
     1,PF2,DS,PHTW,PHSW,SWPOW,BETA)                                         0050
C     END OF PACKING SPECS **************************************************
```

MAINLINE PACKED

```
C     START OF PHYSICAL PROP SPECS ***********************************************
      NRHOM=0                                                                    0051
      IF(MCAN.EQ.1) WRITE(NTP,5032)                                              0052
 5032 FORMAT(/T2,'COMPONENT-DENSITY DATA MUST BE AVAILABLE AS DESCRIBED          0053
     1IN THE',/T2,'DISTILLATION PACKAGE MANUAL,SECTION 3.4',/T2,'IS SOLU         0054
     2TION-DENSITY DATA ALSO AVAILABLE?')                                        0055
      READ(NCR,2) RONAME                                                         0056
      IF(RONAME.EQ.'NO') GO TO 531                                               0057
      NRHOM=1                                                                    0058
  531 CONTINUE                                                                   0059
      IF(MCAN.EQ.1) WRITE(NTP,5033)                                              0060
 5033 FORMAT(/T2,'COMPOUND CLASSES ARE:',/T2,'(1) ALCOHOLS AND WATER',/T         0061
     12,'(2) HYDROCARBONS (SAT.,UNSAT. AND AROM.) AND ETHERS',/T2,'(3) O         0062
     2THER ORGANICS',/T2,'(4) INORGANIC COMPOUNDS',/T2,'ENTER CLASS NO.          0063
     3INTO WHICH THE LIGHT AND HEAVY COMPONENTS FALL',/T2,'EG. FOR WATER         0064
     4 AND NITRIC ACID ENTER 1,4')                                               0065
      READ(NCR,1) NLCAT,NHCAT                                                    0066
C
C     START OF PACKING HEIGHT CALCULATIONS ***************************************
C     TO '302 CONTINUE' SEARCHES FOR POINT OF MAXIMUM LIQUID FLOW
C     IN COLUMN. RELEVANT FLOW AND TEMPERATURE DATA AT THIS POINT
C     ARE NOTED.
C     SUBSEQUENT CALCULATIONS ARE BASED ON THESE DATA.
      FLOR=0.0                                                                   0067
      DO 31 N=NTOT,MPT1,-1                                                       0068
      FLO1=LRWT(N)                                                               0069
      IF(FLOR.GT.FLO1) GO TO 31                                                  0070
      NFLOR=N                                                                    0071
      FLOR=FLO1                                                                  0072
   31 CONTINUE                                                                   0073
```

MAINLINE PACKED

```
      IF(MFIN.GT.999) GO TO 301                                          0074
      FLOS=0.0                                                           0075
      DO 32 M=MPTS,MFIN,-1                                               0076
      FLO2=LSWT(M)                                                       0077
      IF(FLOS.GT.FLO2) GO TO 32                                          0078
      NFLOS=M                                                            0079
      FLOS=FLO2                                                          0080
   32 CONTINUE                                                           0081
      IF(FLOR.GT.FLOS) GO TO 301                                         0082
      FLOL=LSWT(NFLOS)                                                   0083
      FLOV=VSWT(NFLOS)                                                   0084
      FVAP=VAPS(NFLOS)                                                   0085
      TFLOL=TSL(NFLOS)                                                   0086
      TFLOV=TSV(NFLOS)                                                   0087
      XLIQ=XSOP(NFLOS)                                                   0088
      YVAP=YSOP(NFLOS)                                                   0089
      NPOINT=NFLOS                                                       0090
      GO TO 302                                                          0091
  301 CONTINUE                                                           0092
      FLOL=LRWT(NFLOR)                                                   0093
      FLOV=VRWT(NFLOR)                                                   0094
      FVAP=VAPR(NFLOR)                                                   0095
      TFLOL=TRL(NFLOR)                                                   0096
      TFLOV=TRV(NFLOR)                                                   0097
      XLIQ=XROP(NFLOR-MPTS)                                              0098
      YVAP=YROP(NFLOR-MPTS)                                              0099
      NPOINT=NFLOR                                                       0100
  302 CONTINUE                                                           0101
C     DENSITY OF LIQUID PHASE AT TEMPERATURE 'TFLOL',
      NPHYS=1                                                            0102
      CALL RHOLIQ(NPHYS,NRHOM,NLCAT,NHCAT,TFLOL,XLIQ,RHOL1,RHOL2,RHOL)   0103
      RHOL=RHOL*RHOCON                                                   0104
      RHOV=FLOV*492./(FVAP*359.*(459.4+(TFLOV*1.8+32.)))                 0105
      RHOFAC=(RHOV/RHOL)**0.5                                            0106
```

MAINLINE PACKED

```
C     ABCIS IS THE X-AXIS COORDINATE ON THE U.S. STONEWARE FLOODING
C     AND LOADING CURVE DIAGRAM.
      ABCIS=FLOL*RHOFAC/FLOV                                              0107
      WRITE(NTP,4401)                                                     0108
 4401 FORMAT(/T2,'DO YOU WANT TO SPECIFY THE COLUMN DIAMETER?')           0109
      READ(NCR,2) DNAME                                                   0110
      IF(DNAME.NE.'YES') GO TO 441                                        0111
      WRITE(NTP,4411)                                                     0112
 4411 FORMAT(T2,'ENTER COLUMN DIAMETER IN METRES.')                       0113
      READ(NCR,1) DIAM                                                    0114
      IF(DIAM.GT.2) GO TO 902                                             0115
  444 CONTINUE                                                            0116
      DM1=DIAM                                                            0117
      VPDL=-1.                                                            0118
      AREA=0.7854*(DIAM**2.)                                              0119
      VTO=FLOV/AREA                                                       0120
      ALTO=FLOL/AREA                                                      0121
      VLOAD=1.05*VTO                                                      0122
      GO TO 442                                                           0123
  441 CONTINUE                                                            0124
      WRITE(NTP,5040)                                                     0125
 5040 FORMAT(/T2,'SPECIFY VAPOUR PRESSURE DROP AT LOADING FROM FOLLOWING  0126
     1:',/T2,'(1) 0.041,(2) 0.082,(3) 0.204,(4) 0.408,(5) 0.816,(6) 1.22  0127
     24 (KPA/M)',/T2,'NOTE: 0.408 (CODE 4) REPRESENTS APPROX. 60% OF FLO  0128
     3ODING VALUE.')                                                      0129
      WRITE(NTP,5041)                                                     0130
 5041 FORMAT(T2,'ENTER APPROPRIATE CODE NO.')                             0131
      READ(NCR,1) NFLOOD                                                  0132
      NPHYS=1                                                             0133
C     'GO TO' LOOP 410 SEARCHES FOR ACCEPTABLE VAPOUR MASS-FLOWRATE
C     AND COLUMN DIAMETER.
  410 CONTINUE                                                            0134
      IF(NFLOOD.EQ.1.OR.NFLOOD.EQ.2) VPDL=NFLOOD*0.041                    0135
      IF(NFLOOD.EQ.3.OR.NFLOOD.EQ.4) VPDL=(NFLOOD-2)*0.204                0136
      IF(NFLOOD.EQ.5.OR.NFLOOD.EQ.6) VPDL=(NFLOOD-3)*0.408                0137
```

MAINLINE PACKED

```
C      'FDATA' FITS AN N'TH ORDER POLYNOMIAL TO THE LOADING CURVE
C      CORRESPONDING TO THE SELECTED PRESSURE DROP SPECIFICATIONS.
C      Y-AXIS COORDINATE (ORD) CORRESPONDING TO X-AXIS COORDINATE
C      (ABCIS) IS RETURNED.
       CALL FDATA(NFLOOD,ABCIS,ORD)                                          0138
       WRITE(NTP,4001) ABCIS,ORD                                             0139
 4001  FORMAT(/T2,'OPERATING POINT ON SELECTED LOADING CURVE:'/T2,           0140
      1'ABSCISSA=',F9.3,/T2,'ORDINATE=',F9,3)                                0141
  442  CONTINUE                                                              0142
C      LIQUID VISCOSITY CALCULATION.
       CALL VISCL(NPHYS,NLCAT,NHCAT,TFLOL,XLIQ,AMU1,AMU2,AMUML)              0143
C      DENSITY OF WATER AT TEMPERATURE 'TFLOL'.
       CALL RHOH2O(NPHYS,TFLOL,RHOW)                                         0144
       RHOW=RHOW*RHOCON                                                      0145
       IF(VPDL.GT.0.0) GO TO 443                                             0146
       VDASH=VLOAD/ALBKGH                                                    0147
       ORD=((VDASH**2.)*CF*(AMUML**0.2)*(RHOW/RHOL))/(GC*RHOL*RHOV)          0148
       NPHYS=2                                                               0149
       WRITE(NTP,4003) ABCIS,ORD                                             0150
 4003  FORMAT(/T2,'OPERATING POINT COORDINATES:'/T2,'ABSCISSA=',F9.3,        0151
      1/T2,'ORDINATE=',F9.3,/T2,'IS THIS POINT WITHIN LOADING REGION?')      0152
       READ(NCR,2) PNAME                                                     0153
       IF(PNAME.EQ.'YES') GO TO 400                                          0154
       WRITE(NTP,4402)                                                       0155
 4402  FORMAT(T2,'ENTER ANOTHER DIAMETER.')                                  0156
       READ(NCR,1) DIAM                                                      0157
       IF(DIAM.GT.2) GO TO 902                                               0158
       GO TO 444                                                             0159
  443  CONTINUE                                                              0160
C      CALCULATION OF VAPOUR MASS-FLOWRATE AT LOADING.
       VDASH=((ORD*GC*RHOV*RHOL)/(CF*(AMUML**0.2)*(RHOW/RHOL)))**0.5         0161
       VLOAD=VDASH*ALBKGH                                                    0162
```

MAINLINE PACKED

```
      DDIAM=0.01                                                        0163
      DIAM=2.0                                                          0164
C     SEARCH FOR DIAMETER THAT WILL RESULT IN A VAPOUR FLOWRATE
C     NOT GREATER THAN THE CALCULATED LOADING VALUE.
      DO 40 K=1,200                                                     0165
      DIAM=DIAM-DDIAM                                                   0166
      CSAREA=0.7854*(DIAM**2.)                                          0167
      VTEST=FLOV/CSAREA                                                 0168
      VDIFF=VLOAD-VTEST                                                 0169
      IF(K.EQ.1.AND.VDIFF.LE.0.0001) GO TO 902                          0170
      IF(VDIFF.LE.0.0001) GO TO 400                                     0171
      DMI=DIAM                                                          0172
      AREA=CSAREA                                                       0173
      VTO=VTEST                                                         0174
      ALTO=FLOL/AREA                                                    0175
   40 CONTINUE                                                          0176
  400 CONTINUE                                                          0177
C     PRINTOUT OF RECOMMENDED TOWER DIAMETER.AND RESULTING FLOWRATES.
      WRITE(NTP,4002) VLOAD,DMI,VTO,ALTO                                0178
 4002 FORMAT(/T2,'FOR GIVEN VAPOUR PRESSUR-DROP:',/T2,'SUPERFICIAL GAS V 0179
     1ELOCITY=',F8.2,2X,'KG/HR.SQM',/T2,'RECOMMENDED TOWER DIAM.=',F9.2,  0180
     22X,'M',/T2,'RESULTANT SUP. GAS VEL.=',F9.2,2X,'KG/HR.SQM',/T2,'RES 0181
     3ULTANT SUP. LIQ VEL.=',F9.2,2X,'KG/HR.SQM',/T2,'ARE THESE FIGURES  0182
     4SATISFACTORY?')                                                   0183
      READ(NCR,2) CHNAME                                                0184
      IF(CHNAME.EQ.'YES') GO TO 401                                     0185
C     IF ABOVE NOT ACCEPTABLE'REQUEST NEW P.D. AND REPEAT CALCS.
      WRITE(NTP,5042)                                                   0186
 5042 FORMAT(/T2,'SPECIFY VAPOUR PRESSURE DROP AT LOADING FROM FOLLOWING 0187
     1:',/T2,'(1) 0.041,(2) 0.082,(3) 0.204,(4) 0.408,(5) 0.816,(6) 1.22  0188
     24 (KPA/M)')                                                       0189
      READ(NCR,1) NFLOOD                                                0190
      NPHYS=2                                                           0191
      GO TO 410                                                         0192
  401 CONTINUE                                                          0193
```

MAINLINE PACKED

```
C
C       COLUMN DIAMETER AND MASS FLOWS/UNIT AREA ARE NOW ESTABLISHED.
C       MASS TRANSFER COEFFICIENTS ARE NOW CALCULATED.
        NPHYS=1                                                          0194
C       LOOP 41 CALCULATES THE MASS TRANSFER COEFFICIENTS IN THE STRIPPING
C       SECTION OF THE COLUMN,USING SHULMAN'S PROCEDURE,
C       EQUATIONS AND LIMITATIONS USED HERE WERE FIRST PROPOSED BY
C       SHULMAN. A DETAILED DISCUSSION CAN BE FOUND IN TREYBAL,R.E.,
C       'MASS TRANSFER OPERATIONS',2ND ED.,CHAPTER 6,OR IN THE
C       DISTILLATION PACKAGE MANUAL,SECTION 3.
        IF(MFIN.GT.999) MFIN=MPTS                                        0195
        DO 41 M=MPTS,MFIN,-1                                             0196
        VT1=VSWT(M)/(AREA*ALBKGH)                                        0197
        ALT1=LSWT(M)/(AREA*ALBKGH)                                       0198
        TEMPV=TSV(M)                                                     0199
        TEMPL=TSL(M)                                                     0200
        XLIQ=XSOP(M)                                                     0201
        NPH=2                                                            0202
        CALL RHOLIQ(NPH,NRHOM,NLCAT,NHCAT,TEMPL,XLIQ,RHOL1,RHOL2,RHOL)   0203
        RHOL=RHOL*RHOCON                                                 0204
        RHOV=VSWT(M)*492./(VAPS(M)*359.*(459.4+(TEMPV*1.8+32.)))         0205
        CALL VISCL(NPH,NLCAT,NHCAT,TEMPL,XLIQ,AMU1,AMU2,AMUML)           0206
        IF(ALT1.GT.1500.) GO TO 411                                      0207
        AM=AMF1                                                          0208
        ANFUNC=ANF1A*(1./(10.**4))*ALT1+ANF1B                            0209
        P=PF1                                                            0210
        GO TO 412                                                        0211
  411   CONTINUE                                                         0212
        AM=AMF2                                                          0213
        ANFUNC=ANF2A*(1./(10.**4))*ALT1+ANF2B                            0214
        P=PF2                                                            0215
  412   CONTINUE                                                         0216
        IF(ABS(RHOV-0.075).LT.0.002) GO TO 413                           0217
        VT1=VT1*((0.075/RHOV)**0.5)                                      0218
  413   CONTINUE                                                         0219
        AAW=AM*(VT1**ANFUNC)*(ALT1**P)                                   0220
```

MAINLINE PACKED

```
      XST=XSOP(M)                                                       0221
C     SURFACE TENSION CALCULATION.
      CALL SURTEN(NPHYS,NLCAT,NHCAT,TEMPL,XST,SIGMA)                    0222
      SIGMA=SIGMA*SIGCON                                                0223
      IF(NTYPE.EQ.2) GO TO 414                                          0224
      PFACT1=6.85*(1./(10.**5))*(AMUML**0.02)*(SIGMA**0.99)             0225
      PFACT2=(DS**1.21)*(RHOL**0.37)                                    0226
      GO TO 415                                                         0227
  414 CONTINUE                                                          0228
      PFACT1=1.641*(1./(10.**4))*(AMUML**0.04)*(SIGMA**0.55)            0229
      PFACT2=(DS**1.56)*(RHOL**0.37)                                    0230
  415 CONTINUE                                                          0231
      PHIS=PFACT1/PFACT2                                                0232
      PHISW=PHSW/(DS**SWPOW)                                            0233
      PHITW=PHTW*(1./(10.**5))*(ALT1**BETA)/(DS**2.)                    0234
      IF(NTYPE.EQ.2) GO TO 417                                          0235
      AFACT2=(RHOL**0.84)*(0.1183*(ALT1**0.43)-1.)                      0236
      AFACT3=(SIGMA/73.)**(0.925-0.262*ALOG10(ALT1))                    0237
      IF(AMUML.GT.12.) GO TO 416                                        0238
      AFACT1=0.897*(ALT1**0.57)*(AMUML**0.13)                           0239
      GO TO 419                                                         0240
  416 CONTINUE                                                          0241
      AFACT1=0.575*(ALT1**0.57)*(AMUML**0.31)                           0242
      GO TO 419                                                         0243
  417 CONTINUE                                                          0244
      AFACT2=(RHOL**0.84)*(0.212*(ALT1**0.413)-1.)                      0245
      AFACT3=(SIGMA/73.)**(1.033-0.262*ALOG10(ALT1))                    0246
      IF(AMUML.GT.20.) GO TO 418                                        0247
      AFACT1=1.291*(ALT1**0.57)*(AMUML**0.13)                           0248
      GO TO 419                                                         0249
  418 CONTINUE                                                          0250
      AFACT1=0.752*(ALT1**0.57)*(AMUML**0.31)                           0251
  419 CONTINUE                                                          0252
      HFUNC=AFACT1*AFACT3/AFACT2                                        0253
      PHIOW=PHITW-PHISW                                                 0254
      PHIO=PHIOW*HFUNC                                                  0255
      PHIT=PHIO+PHIS                                                    0256
      AA=AAW*PHIO/PHIOW                                                 0257
      EO=E-PHIT                                                         0258

                                                        MAINLINE PACKED
```

```
C       VAPOUR DIFFUSIVITY CALCULATION.
        CALL DIFVAP(NPHYS,PT,TEMPV,DIFUSV)                                  0259
        DIFUSV=DIFUSV*DVCON                                                 0260
        YMU=YSOP(M)                                                         0261
C       VAPOUR VISCOSITY CALCULATION.
        CALL VISCV(NPHYS,TEMPV,YMU,AMUMV)                                   0262
        AMUMV=AMUMV*AMVCON                                                  0263
        SCHV=AMUMV/(RHOV*DIFUSV)                                            0264
        VMOLE=VAPS(M)/AREA                                                  0265
        FA1=((AMUMV*(1.-EO)/(DS*VTI))**0.36)                                0266
        FA2=(VMOLE*1.195/(SCHV**(2./3.)))                                   0267
        FV=FA1*FA2                                                          0268
C       VAPOUR MASS TRANSFER COEFFICIENT.
        KY(M)=FV                                                            0269
        XDIF=XSOP(M)                                                        0270
C       LIQUID DIFFUSIVITY CALCULATION.
        CALL DIFLIQ(NPHYS,NLCAT,NHCAT,TEMPL,XDIF,DIFUSL)                    0271
        DIFUSL=DIFUSL*DLCON                                                 0272
        XRHO=XSOP(M)                                                        0273
        CALL RHOLIQ(NPH,NRHOM,NLCAT,NHCAT,TEMPL,XRHO,RHOL1,RHOL2,RHOL)      0274
        RHOL=RHOL*RHOCON                                                    0275
        AMUML=AMUML*AMLCON                                                  0276
        SCHL=AMUML/(RHOL*DIFUSL)                                            0277
        FA1=(DS*ALTI/AMUML)**0.45                                           0278
        FA2=(DIFUSL/DS)*25.1*(SCHL**0.5)                                    0279
        NR=0                                                                0280
        CALL RHOLIQ(NPH,NR,NLCAT,NHCAT,TEMPL,XRHO,RHOL1,RHOL2,RHOL)         0281
        CRHO=RHOL2*RHOCON/MW(2)                                             0282
        FL=FA1*FA2*CRHO                                                     0283
C       LIQUID MASS TRANSFER COEFFICIENT.
        KX(M)=FL*ALBKGH                                                     0284
        L(M)=LIQS(M)/AREA                                                   0285
        V(M)=VMOLE                                                          0286
        ASURF(M)=AA/ASFSM                                                   0287
        NPHYS=2                                                             0288
   41   CONTINUE                                                            0289
        IF(MFIN.EQ.MPTS) MFIN=1000                                          0290
```

MAINLINE PACKED

```
C
C       LOOP 42 CALCULATES THE MASS TRANSFER COEFFICIENTS IN THE RECTIFYING
C       SECTION OF THE COLUMN, USING SHULMAN'S PROCEDURE.
C       COMPUTATIONAL PROCEDURE IDENTICAL TO LOOP 41.
        DO 42 N=NTOT,MPT1,-1                                                    0291
        NPHYS=2                                                                 0292
        NRS=N-MPTS                                                              0293
        VT1=VRWT(N)/(AREA*ALBKGH)                                               0294
        ALT1=LRWT(N)/(AREA*ALBKGH)                                              0295
        TEMPV=TRV(N)                                                            0296
        TEMPL=TRL(N)                                                            0297
        XLIQ=XROP(NRS)                                                          0298
        CALL RHOLIQ(NPHYS,NRHOM,NLCAT,NHCAT,TEMPL,XLIQ,RHOL1,RHOL2,RHOL)        0299
        RHOL=RHOL*RHOCON                                                        0300
        RHOV=VRWT(N)*492./(VAPR(N)*359.*(459.4+(TEMPV*1.8+32.)))                0301
        CALL VISCL(NPHYS,NLCAT,NHCAT,TEMPL,XLIQ,AMU1,AMU2,AMUML)                0302
        IF(ALT1.GT.1500.) GO TO 421                                             0303
        AM=AMF1                                                                 0304
        ANFUNC=ANF1A*(1./(10.**4))*ALT1+ANF1B                                   0305
        P=PF1                                                                   0306
        GO TO 422                                                               0307
  421   CONTINUE                                                                0308
        AM=AMF2                                                                 0309
        ANFUNC=ANF2A*(1./(10.**4))*ALT1+ANF2B                                   0310
        P=PF2                                                                   0311
  422   CONTINUE                                                                0312
        IF(ABS(RHOV-0.075).LT.0.002) GO TO 423                                  0313
        VT1=VT1*((0.075/RHOV)**0.5)                                             0314
  423   CONTINUE                                                                0315
        AAW=AM*(VT1**ANFUNC)*(ALT1**P)                                          0316
        XST=XROP(NRS)                                                           0317
        CALL SURTEN(NPHYS,NLCAT,NHCAT,TEMPL,XST,SIGMA)                          0318
        SIGMA=SIGMA*SIGCON                                                      0319

                                                                    MAINLINE PACKED
```

```
      IF(NTYPE.EQ.2) GO TO 424                                          0320
      PFACT1=6.85*(1./(10.**5))*(AMUML**0.02)*(SIGMA**0.99)             0321
      PFACT2=(DS**1.21)*(RHOL**0.37)                                    0322
      GO TO 425                                                         0323
  424 CONTINUE                                                          0324
      PFACT1=1.641*(1./(10.**4))*(AMUML**0.04)*(SIGMA**0.55)            0325
      PFACT2=(DS**1.56)*(RHOL**0.37)                                    0326
  425 CONTINUE                                                          0327
      PHIS=PFACT1/PFACT2                                                0328
      PHISW=PHSW/(DS**SWPOW)                                            0329
      PHITW=PHTW*(1./(10.**5))*(ALT1**BETA)/(DS**2.)                    0330
      IF(NTYPE.EQ.2) GO TO 427                                          0331
      AFACT2=(RHOL**0.84)*(0.1183*(ALT1**0.43)-1.)                      0332
      AFACT3=(SIGMA/73.)**(0.925-0.262*ALOG10(ALT1))                    0333
      IF(AMUML.GT.12.) GO TO 426                                        0334
      AFACT1=0.897*(ALT1**0.57)*(AMUML**0.13)                           0335
      GO TO 429                                                         0336
  426 CONTINUE                                                          0337
      AFACT1=0.575*(ALT1**0.57)*(AMUML**0.31)                           0338
      GO TO 429                                                         0339
  427 CONTINUE                                                          0340
      AFACT2=(RHOL**0.84)*(0.212*(ALT1**0.413)-1.)                      0341
      AFACT3=(SIGMA/73.)**(1.033-0.262*ALOG10(ALT1))                    0342
      IF(AMUML.GT.20.) GO TO 428                                        0343
      AFACT1=1.291*(ALT1**0.57)*(AMUML**0.13)                           0344
      GO TO 429                                                         0345
  428 CONTINUE                                                          0346
      AFACT1=0.752*(ALT1**0.57)*(AMUML**0.31)                           0347
  429 CONTINUE                                                          0348
      HFUNC=AFACT1*AFACT3/AFACT2                                        0349
      PHIOW=PHITW-PHISW                                                 0350
      PHIO=PHIOW*HFUNC                                                  0351
      PHIT=PHIO+PHIS                                                    0352
      AA=AAW*PHIO/PHIOW                                                 0353
      EO=E-PHIT                                                         0354
```

MAINLINE PACKED

```
      CALL DIFVAP(NPHYS,PT,TEMPV,DIFUSV)                                0355
      DIFUSV=DIFUSV*DVCON                                               0356
      YMU=YROP(NRS)                                                     0357
      CALL VISCV(NPHYS,TEMPV,YMU,AMUMV)                                 0358
      AMUMV=AMUMV*AMVCON                                                0359
      SCHV=AMUMV/(RHOV*DIFUSV)                                          0360
      VMOLE=VAPR(N)/AREA                                                0361
      FA1=((AMUMV*(1.-EO)/(DS*VTI))**0.36)                              0362
      FA2=(VMOLE*1.195/(SCHV**(2./3.)))                                 0363
      FV=FA1*FA2                                                        0364
      KY(N)=FV                                                          0365
      XDIF=XROP(NRS)                                                    0366
      CALL DIFLIQ(NPHYS,NLCAT,NHCAT,TEMPL,XDIF,DIFUSL)                  0367
      DIFUSL=DIFUSL*DLCON                                               0368
      XRHO=XROP(NRS)                                                    0369
      CALL RHOLIQ(NPHYS,NRHOM,NLCAT,NHCAT,TEMPL,XRHO,RHOL1,RHOL2,RHOL)  0370
      RHOL=RHOL*RHOCON                                                  0371
      AMUML=AMUML*AMLCON                                                0372
      SCHL=AMUML/(RHOL*DIFUSL)                                          0373
      FA1=(DS*ALTI/AMUML)**0.45                                         0374
      FA2=(DIFUSL/DS)*25.1*(SCHL**0.5)                                  0375
      NR=0                                                              0376
      CALL RHOLIQ(NPHYS,NR,NLCAT,NHCAT,TEMPL,XRHO,RHOL1,RHOL2,RHOL)     0377
      CRHO=RHOL2*RHOCON/MW(2)                                           0378
      FL=FA1*FA2*CRHO                                                   0379
      KX(N)=FL*ALBKGH                                                   0380
      L(N)=LIQR(N)/AREA                                                 0381
      V(N)=VMOLE                                                        0382
      ASURF(N)=AA/ASFSM                                                 0383
   42 CONTINUE                                                          0384

                                                        MAINLINE PACKED
```

```
C
C     PREPARATION FOR HEIGHT_OF_PACKING INTEGRATION.
C
      DO 43 N=NTOT,MPT1,-1                                              0385
      NRS=N-MPTS                                                        0386
      SLOPK(N)=-KX(N)/KY(N)                                             0387
      CFAK(N)=YROP(NRS)-SLOPK(N)*XROP(NRS)                              0388
   43 CONTINUE                                                          0389
      IF(MFIN.GT.999) GO TO 403                                         0390
      DO 44 M=MPTS,MFIN,-1                                              0391
      SLOPK(M)=-KX(M)/KY(M)                                             0392
      CFAK(M)=YSOP(M)-SLOPK(M)*XSOP(M)                                  0393
   44 CONTINUE                                                          0394
  403 CONTINUE                                                          0395
      DO 45 N=NTOT,MPT1,-1                                              0396
      NRS=N-MPTS                                                        0397
      MN=1                                                              0398
      DELX=0.001                                                        0399
      XG=XROP(NRS)-DELX                                                 0400
      SL=SLOPK(N)                                                       0401
      CFAC=CFAK(N)                                                      0402
  451 CONTINUE                                                          0403
      CALL XYALFA(ALFA,ALFCON,ALF,MALF,NDATA,XALF,XG)                   0404
      YG=ALF*XG/(1.+XG*(ALF-1.))                                        0405
      YTEST=SL*XG+CFAC                                                  0406
      IF(ABS(YG-YTEST).LT.0.0001) GO TO 452                             0407
      NCALL=1                                                           0408
      CALL SEARCH(YG,YTEST,XG,DELX,MN,NCALL)                            0409
      GO TO 451                                                         0410
  452 CONTINUE                                                          0411
      IF(XG.GE.XROP(NRS).OR.YG.LE.YROP(NRS)) GO TO 901                  0412
      XI(N)=XG                                                          0413
      YI(N)=YG                                                          0414
   45 CONTINUE                                                          0415
      IF(MFIN.GT.999) GO TO 404                                         0416
```

MAINLINE PACKED

```
      DO 46 M=MPTS,MFIN,-1                                              0417
      MN=1                                                              0418
      DELX=0.001                                                        0419
      XG=XSOP(M)-DELX                                                   0420
      SL=SLOPK(M)                                                       0421
      CFAC=CFAK(M)                                                      0422
  461 CONTINUE                                                          0423
      CALL XYALFA(ALFA,ALFCON,ALF,MALF,NDATA,XALF,XG)                   0424
      YG=ALF*XG/(1.+XG*(ALF-1.))                                        0425
      YTEST=SL*XG+CFAC                                                  0426
      IF(ABS(YG-YTEST).LT.0.0001) GO TO 462                             0427
      NCALL=1                                                           0428
      CALL SEARCH(YG,YTEST,XG,DELX,MN,NCALL)                            0429
      GO TO 461                                                         0430
  462 CONTINUE                                                          0431
      IF(XG.GE.XSOP(M).OR.YG.LE.YSOP(M)) GO TO 901                      0432
      XI(M)=XG                                                          0433
      YI(M)=YG                                                          0434
   46 CONTINUE                                                          0435
  404 CONTINUE                                                          0436
C
C     CALCULATION OF VALUES OF THE FUNCTION TO BE INTEGRATED.
C     (RECTIFYING SECTION)
      IF(KX(NPOINT).LT.KY(NPOINT)) GO TO 611                            0437
C     'DO' LOOP 61: PRINCIPAL MASS TRANSFER RESISTANCE IN VAPOUR PHASE.
      NSORT=1                                                           0438
      DO 61 N=NPTS,NFIN,-1                                              0439
      NRS=N+MPTS                                                        0440
      VYR(N)=V(NRS)*YROP(N)                                             0441
      FUNCR(N)=1./(KY(NRS)*ASURF(NRS)*(YI(NRS)-YROP(N)))                0442
   61 CONTINUE                                                          0443
      GO TO 612                                                         0444
  611 CONTINUE                                                          0445
```

MAINLINE PACKED

```
C     'DO' LOOP 62: PRINCIPAL MASS TRANSFER RESISTANCE IN LIQUID PHASE.
      NSORT=2                                                          0446
      DO 62 N=NPTS,NFIN,-1                                             0447
      NRS=N+MPTS                                                       0448
      VYR(N)=L(NRS)*XROP(N)                                            0449
      FUNCR(N)=1./(KX(NRS)*ASURF(NRS)*(XROP(N)-XI(NRS)))               0450
   62 CONTINUE                                                         0451
  612 CONTINUE                                                         0452
      TOP=VYR(NPTS)                                                    0453
      FEED=VYR(1)                                                      0454
      NPTI=NPTS-NFIN+1                                                 0455
C     INTEGRATION OVER RECTIFYING SECTION.
      CALL SIMPS(TOP,FEED,NPTI,FUNCR,SUM)                              0456
C     HEIGHT OF PACKING FOR RECTIFYING SECTION (M).
      ZRECT=SUM                                                        0457
      IF(NFIN.GT.999) GO TO 699                                        0458
C     CALCULATION OF VALUES OF THE FUNCTION TO BE INTEGRATED.
C     (STRIPPING SECTION)
      IF(NSORT.EQ.2) GO TO 613                                         0459
C     'DO' LOOP 63: PRINCIPAL MASS TRANSFER RESISTANCE IN VAPOUR PHASE.
      DO 63 M=MPTS,NFIN,-1                                             0460
      VYS(M)=V(M)*YSOP(M)                                              0461
      FUNCS(M)=1./(KY(M)*ASURF(M)*(YI(M)-YSOP(M)))                     0462
   63 CONTINUE                                                         0463
      GO TO 614                                                        0464
  613 CONTINUE                                                         0465
C     'DO' LOOP 64: PRINCIPAL MASS TRANSFER RESISTANCE IN LIQUID PHASE.
      DO 64 M=MPTS,NFIN,-1                                             0466
      VYS(M)=L(M)*XSOP(M)                                              0467
      FUNCS(M)=1./(KX(M)*ASURF(M)*(XSOP(M)-XI(M)))                     0468
   64 CONTINUE                                                         0469
  614 CONTINUE                                                         0470
      FEED=VYS(MPTS)                                                   0471
      BOT=VYS(NFIN)                                                    0472
C     INTEGRATION OVER STRIPPING SECTION.
      MPTI=MPTS-NFIN+1                                                 0473
      CALL SIMPS(FEED,BOT,MPTI,FUNCS,SUM)                              0474
```

MAINLINE PACKED

```
C     HEIGHT OF PACKING FOR STRIPPING SECTION (M).
      ZSTRIP=SUM                                                        0475
  699 CONTINUE                                                          0476
      CALL PACRES(NDAT,NPTS,MPTS,NFIN,MFIN,NTYPE,NPACK,DM1,ZRECT,ZSTRIP,0477
     1VPDL,ASURF,FUNCR,FUNCS,L,V,VYR,VYS,XI,YI,KX,KY,NSORT)             0478
      GO TO 999                                                         0479
C     ERROR MESSAGES ************************************************
  901 CONTINUE                                                          0480
      WRITE(NTP,9001) XG,YG                                             0481
 9001 FORMAT(/T2,'PINCH POINT OCCURS AT POINT',F7.4,',',F6.4,' ON THE', 0482
     1/T2,'EQUILIBRIUM CURVE.',/T2,'THE REFLUX RATIO OR THE DISTILLATE A0483
     2ND BOTTOMS COMPOSITIONS',/T2,'SHOULD BE ADJUSTED ACCORDINGLY.')   0484
      WRITE(NTP,9011)                                                   0485
 9011 FORMAT(/T2,'EQUILIBRIUM AND OPERATING LINE DATA WILL BE PRINTED OU0486
     1T.',/T2,'THIS SHOULD BE PLOTTED ON AN X-Y DIAGRAM,AND THE SYSTEM',0487
     2/T2,'CHECKED FOR AZEOTROPISM.')                                   0488
 9021 FORMAT(T5,F6.4,2X,F6.4)                                           0489
 9031 FORMAT(/T2,'EQUILIBRIUM DATA:',/T8,'X',7X,'Y')                    0490
 9041 FORMAT(/T2,'POINTS ON OPERATING LINES:',/T8,'X',7X,'Y')           0491
 9051 FORMAT(T2,'POINT OF INTERSECTION OF OPERATING LINES:',F7.4,       0492
     12X,F6.4)                                                          0493
      WRITE(NTP,9031)                                                   0494
      WRITE(NTP,9021) (XEQUIL(K),YEQUIL(K),K=1,NDAT,5)                  0495
      WRITE(NTP,9041)                                                   0496
      WRITE(NTP,9021) (XSOP(M),YSOP(M),M=1,MPTS,5)                      0497
      WRITE(NTP,9051) XCR,YCR                                           0498
      WRITE(NTP,9021) (XROP(N),YROP(N),N=1,NPTS,5)                      0499
      GO TO 999                                                         0500
  902 CONTINUE                                                          0501
      WRITE(NTP,9002)                                                   0502
 9002 FORMAT(/T2,'DIAMETER OF THE COLUMN EXCEEDS 2 METRES.',            0503
     1/T2,'SYSTEM UNSUITABLE FOR PACKED COLUMN DISTILLATION.')          0504
  999 CONTINUE                                                          0505
      WRITE(NTP,9999)                                                   0506
 9999 FORMAT(/T2,'END OF PROGRAM.')                                     0507
      END                                                               0508

                                                          MAINLINE PACKED
```

```
      SUBROUTINE PACRES(NDAT,NPTS,MPTS,NFIN,MFIN,NTYPE,NPACK,DIAMM,ZRECT     0001
     1,ZSTRIP,VPDL,ASURF,FUR,FUS,L,V,VYR,VYS,XI,YI,KX,KY,NSORT)              0002
C     ROUTINE FOR PRINTING RESULTS FOR DESIGN OF PACKED COLUMNS.
C     **********************************
C     INSERT COMMON BLOCK 'BINCOM' HERE.
C     **********************************
      DIMENSION ASURF(1),FUR(1),FUS(1),KX(1),KY(1),L(1)                      0003
      DIMENSION V(1),VYR(1),VYS(1),XI(1),YI(1)                               0004
      REAL KX,KY,L,LH,LHCONV,LHMW,LIQR,LIQS,LRWT,LSWT,MW                     0005
      NTOT=NPTS+MPTS                                                         0006
      MPTI=MPTS+NFIN                                                         0007
    1 FORMAT(A3)                                                             0008
 1001 FORMAT(/T2,'DESIGN OF A PACKED DISTILLATION COLUMN.',/T2,              0009
     1'MASS TRANSFER COEFFICIENT CALCULATIONS BY SHULMANS METHOD.')          0010
 1002 FORMAT(/T2,'FEED COMPONENTS: (1) ',5A6,/T19,'(2) ',5A6)                0011
 1003 FORMAT(/T2,'FEED STREAM SPECIFICATIONS:',/T4,'FLOWRATE:',F10.2,3X,'    0012
     1'KGMOLE/HR',/T4,'COMPOSITION:',F9.4,1X,'MOLE FRAC',/T4,'QVALUE:',      0013
     2F12.2,/T4,'TEMPERATURE:',F7.2,3X,'DEG C',/T4,'ENTHALPY:',F10.2,        0014
     33X,'KJ/KGMOLE')                                                        0015
 1007 FORMAT(/T2,'DISTILLATE WITHDRAWAL RATE:',F7.2,' KGMOLE/HR',/T2,        0016
     1'DISTILLATE COMPOSITION:',F13.4)                                       0017
 1008 FORMAT(/T2,'BOTTOMS WITHDRAWAL RATE:',F7.2,' KGMOLE/HR',/T2,           0018
     1'BOTTOMS COMPOSITION:',F13.4)                                          0019
 1011 FORMAT(/T2,'SIZE AND TYPE OF PACKING USED:',F6.1,1X,'MM',              0020
     13X,'RASCHIG RINGS')                                                    0021
 1012 FORMAT(/T2,'SIZE AND TYPE OF PACKING USED:',F6.1,1X,'MM',              0022
     13X,'BERL SADDLES')                                                     0023
 1201 FORMAT(/T2,'TOTAL CONDENSER OPERATION.')                               0024
 1202 FORMAT(/T2,'PARTIAL CONDENSER OPERATION.')                             0025
 1203 FORMAT(T2,'CONDENSER PRESSURE:',F9.2,1X,'KPA')                         0026
 1205 FORMAT(T2,'CONDENSER DUTY:',F13.2,1X,'KW')                             0027
 1206 FORMAT(T2,'REBOILER DUTY:',F17.2,1X,'KW')                              0028
 1207 FORMAT(/T2,'REFLUX RATIO:',F11.4)                                      0029
 1208 FORMAT(T2,'MIN REFLUX RATIO:',F7.4)                                    0030
```

```
2010 FORMAT(/T2,'VAPOUR PRESSURE-DROP AT LOADING:',F6.3,1X,'KPA/M')          0031
2011 FORMAT(/T2,'THE COLUMN DIAMETER IS BASED ON CONDITIONS AT THE POIN      0032
     1T IN',/T2,'THE COLUMN WITH THE GREATEST LIQUID FLOWRATE.',             0033
     2/T2,'(SEE TABLE BELOW)')                                              0034
2012 FORMAT(/T2,'RECOMMENDED COLUMN DIAMETER:',F6.3,1X'M')                  0035
2013 FORMAT(/T2,'HEIGHT OF PACKING REQUIRED:',/T4,                          0036
     1'RECTIFYING SECTION:',F7.2,1X,'M',/T4,                                0037
     2'STRIPPING SECTION:',F8.2'1X,'M')                                     0038
3000 FORMAT(/T2,,DO YOU WANT COMPOSITION,TEMPERATURE AND FLOWRATE DATA      0039
     1TABLES?')                                                             0040
3001 FORMAT(/T2,'COMPOSITION,TEMPERATURE AND FLOWRATE PROFILES:')           0041
3010 FORMAT(T4''COMPOSITION',5X,'TEMPERATURE',6X,'FLOWRATES''               0042
     1/T4,'(MOLE FRAC)',7X,'(DEG C)',7X,'(KGMOLE/HR)',/T4,                  0043
     2'LIQ',5X,'VAP',6X,'BUBBLE PT',6X,'LIQ',5X,'VAP')                      0044
3011 FORMAT(T2,F6.4,2X,F6.4,6X,F7.2,4X,F7.2,1X,F7.2)                        0045
3012 FORMAT(T2,''FEED POSITION:')                                           0046
3013 FORMAT(/T2,'SHULMAN MASS TRANSFER CALCULATIONS:',/T4,                  0047
     1'X =LIQUID MOLE FRACTION OF LIGHT COMPONENT',/T4,                     0048
     2'V =VAPOUR RATE IN COLUMN (KGMOLE/HR,SQ M)',/T4,                      0049
     3'A =INTERFACIAL SURFACE AREA OF PACKED SECTION (SQ M/CU M)',/T4,      0050
     4'KY=VAPOUR MASS TRANSFER COEFFICIENT (KGMOLE/HR,SQ M,MOLE FRAC)',     0051
     5/T4,'KX=LIQUID   "       "          "        "  /     " ')           0052
3113 FORMAT(/T2,'SHULMAN MASS TRANSFER CALCULATIONS:',/T4,                  0053
     1'X =LIQUID MOLE FRACTION OF LIGHT COMPONENT',/T4,                     0054
     2'L =LIQUID RATE IN COLUMN (KGMOLE/HR,SQ M)',/T4,                      0055
     3'A =INTERFACIAL SURFACE AREA OF PACKED SECTION (SQ M/CU M)',/T4,      0056
     4'KY=VAPOUR MASS TRANSFER COEFFICIENT (KGMOLE/HR,SQ M,MOLE FRAC)',     0057
     5/T4,'KX=LIQUID   "       "          "        "  /     " ')           0058
3014 FORMAT(/T4,'X',9X,'V',7X,'A',7X,'KY',7X,'KX')                          0059
3114 FORMAT(/T4,'X',9X,'L',7X,'A',7X,'KY',7X,'KX')                          0060
3015 FORMAT(T2,F6.4,F10.3,F8.3,F9.4,F9.4)                                   0061
```

SUBROUTINE PACRES

```
 3016 FORMAT(/T2,'DATA FOR NUMERICAL INTEGRATION TO FIND HEIGHT OF PACKI      0062
     1NG:',/T4,'INTEGRATION OF:  D(Z)=D(V,Y)/KY.A.(YI-Y)',/T4'               0063
     2,Y =VAPOUR MOLE FRACTION OF LIGHT COMPONENT',/T4,                      0064
     3'YI=POINT ON EQUIL CURVE CORRESPONDING TO PT Y ON OP. LINE')           0065
 3116 FORMAT(/T2,'DATA FOR NUMERICAL INTEGRATION TO FIND HEIGHT OF PACKI     0066
     1NG:',/T4,'INTEGRATION OF:  D(Z)=D(L,X)/KX.A.(X-XI)',/T4,               0067
     2'X =LIQUID MOLE FRACTION OF LIGHT COMPONENT',/T4,                      0068
     3'XI=POINT ON EQUIL CURVE CORRESPONDING TO PT X ON OP. LINE')           0069
 3017 FORMAT(/T4,'Y',8X,'YI',5X,'1./KY.A.(YI-Y)',5X,'V.Y')                   0070
 3117 FORMAT(/T4,'X',8X,'XI',5X,'1./KX.A.(X-XI)',5X,'L.X')                   0071
 3018 FORMAT(T2,F6.4,3X,F6.4,6X,F7.3,7X,F7.3)                                0072
 3020 FORMAT(/T2,'DO YOU WANT MASS-TRANSFER DATA TABLES?')                   0073
 3021 FORMAT(/T2,'PRINCIPAL MASS TRANSFER RESISTANCE IN VAPOUR PHASE.')      0074
 3022 FORMAT(/T2,'PRINCIPAL MASS TRANSFER RESISTANCE IN LIQUID PHASE.')      0075
C     RESULT TABLE HEADING,COMPONENT NAMES AND FEED STREAM DATA.
      WRITE(NTP,1001)                                                       0076
      WRITE(NTP,1002) ((TITLES(I,J),J=1,5),I=1,2)                           0077
      WRITE(NTP,1003) FTOT,Z,Q,TF,QF(1)                                     0078
C     CONDENSER SPECIFICATIONS.
      IF(NCOND.EQ.1) WRITE(NTP,1201)                                        0079
      IF(NCOND.EQ.2) WRITE(NTP,1202)                                        0080
      WRITE(NTP,1205) QC                                                    0081
      PTK=PT/7.50061683                                                     0082
      WRITE(NTP,1203) PTK                                                   0083
C     REFLUX CONDITIONS,AND DISTILLATE PRODUCT DATA.
      WRITE(NTP,1207) R                                                     0084
      WRITE(NTP,1208) RM                                                    0085
      WRITE(NTP,1007) DTOT,XD                                               0086
C     BOTTOMS PRODUCT DATA,AND REBOILER SPECIFICATIONS.
      WRITE(NTP,1008) BTOT,XB                                               0087
      WRITE(NTP,1206) QB                                                    0088
C     COLUMN SPECIFICATIONS.
      WRITE(NTP,2010) VPDL                                                  0089
      WRITE(NTP,2011)                                                       0090
      WRITE(NTP,2012) DIAMM                                                 0091
      IF(MFIN.GT.999) ZSTRIP=0.0                                            0092
      WRITE(NTP,2013) ZRECT,ZSTRIP                                          0093
```

```
      SIZE=NPACK*12.7                                                    0094
C     PACKING SPECIFICATIONS.
      IF(NTYPE.EQ.1) WRITE(NTP,1011) SIZE                                0095
      IF(NTYPE.EQ.2) WRITE(NTP,1012) SIZE                                0096
C     COMPOSITION,TEMPERATURE AND FLOWRATE PROFILES.
      WRITE(NTP,3000)                                                    0097
      READ(NCR,1) ANAME                                                  0098
      IF(ANAME.NE.'YES') GO TO 301                                       0099
      WRITE(NTP,3001)                                                    0100
      WRITE(NTP,3010)                                                    0101
C     RECTIFYING SECTION.
      WRITE(NTP,3011) (XROP(I-MPTS),YROP(I-MPTS),TRL(I),LIQR(I),VAPR(I),  0102
     1I=NTOT,MPT1,-5)                                                    0103
C     FEED POSITION.
      WRITE(NTP,3012)                                                    0104
C     STRIPPING SECTION.
      IF(MFIN.LT.999) WRITE(NTP,3011) (XSOP(I),YSOP(I),TSL(I),LIQS(I),VA   0105
     1PS(I),I=MPTS,MFIN,-5)                                              0106
  301 CONTINUE                                                           0107
      WRITE(NTP,3020)                                                    0108
      READ(NCR,1) ANAME                                                  0109
      IF(ANAME.NE.'YES') GO TO 999                                       0110
      IF(NSORT.EQ.2) GO TO 310                                           0111
C     MASS-TRANSFER DATA FOR SYSTEMS WITH PRINCIPAL RESISTANCE IN
C     VAPOUR PHASE.
      WRITE(NTP,3021)                                                    0112
      WRITE(NTP,3013)                                                    0113
      WRITE(NTP,3014)                                                    0114
C     RECTIFYING SECTION.
      WRITE(NTP,3015) (XROP(I-MPTS),V(I),ASURF(I),KY(I),KX(I),I=NTOT,MPT   0115
     11,-5)                                                              0116
      WRITE(NTP,3012)                                                    0117
C     STRIPPING SECTION.
      IF(MFIN.LT.999) WRITE(NTP,3015) (XSOP(I),V(I),ASURF(I),KY(I),KX(I)   0118
     1,I=MPTS,MFIN,-5)                                                   0119
      WRITE(NTP,3016)                                                    0120
      WRITE(NTP,3017)                                                    0121
      GO TO 311                                                          0122
```

```
  310 CONTINUE                                                          0123
C     MASS-TRANSFER DATA FOR SYSTEMS WITH PRINCIPAL RESISTANCE IN
C     LIQUID PHASE.
      WRITE(NTP,3022)                                                   0124
      WRITE(NTP,3113)                                                   0125
      WRITE(NTP,3114)                                                   0126
C     RECTIFYING SECTION.
      WRITE(NTP,3015) (XROP(I-MPTS),L(I),ASURF(I),KY(I),KX(I),I=NTOT,MPT 0127
     1I,-5)                                                             0128
      WRITE(NTP,3012)                                                   0129
C     STRIPPING SECTION.
      IF(MFIN.LT.999) WRITE(NTP,3015) (XSOP(I),L(I),ASURF(I),KY(I),KX(I) 0130
     1,I=MPTS,MFIN,-5)                                                  0131
      WRITE(NTP,3116)                                                   0132
      WRITE(NTP,3117)                                                   0133
  311 CONTINUE                                                          0134
C     DATA INTEGRATED TO FIND THE HEIGHT OF PACKING IN EACH SECTION.
C     RECTIFYING SECTION.
      DO 30 I=NTOT,MPT1,-5                                              0135
      IIM=I-MPTS                                                        0136
      IF(NSORT.EQ.1) XY1=YROP(IIM)                                      0137
      IF(NSORT.EQ.1) XY2=YI(I)                                          0138
      IF(NSORT.EQ.2) XY1=XROP(IIM)                                      0139
      IF(NSORT.EQ.2) XY2=XI(I)                                          0140
      WRITE(NTP,3018) XY1,XY2,FUR(IIM),VYR(IIM)                         0141
   30 CONTINUE                                                          0142
      WRITE(NTP,3012)                                                   0143
C     STRIPPING SECTION.
      IF(MFIN.GT.999) GO TO 999                                         0144
      DO 31 I=MPTS,MFIN,-5                                              0145
      IF(NSORT.EQ.1) XY1=YSOP(I)                                        0146
      IF(NSORT.EQ.1) XY2=YI(I)                                          0147
      IF(NSORT.EQ.2) XY1=XSOP(I)                                        0148
      IF(NSORT.EQ.2) XY2=XI(I)                                          0149
      WRITE(NTP,3018) XY1,XY2,FUS(I),VYS(I)                             0150
   31 CONTINUE                                                          0151
  999 CONTINUE                                                          0152
      RETURN                                                            0153
      END                                                               0154
```

```
      SUBROUTINE PACTYP(MCAN,NTYPE,NPACK,NCR,NTP)                       0001
    1 FORMAT( )                                                         0002
    2 FORMAT(A3)                                                        0003
      IF(MCAN.EQ.1) WRITE(NTP,1000)                                     0004
 1000 FORMAT(/T2,'THE PROGRAM DATA PACKAGE INCLUDES PACKING DATA FOR CER 0005
     1AMIC RASCHIG',/T2,'RINGS AND BERL SADDLES.',/T2,'DO YOU WANT TO US 0006
     2E A DIFFERENT TYPE OF PACKING?')                                  0007
      READ(NCR,2) PACTYP                                                0008
      IF(PACTYP.EQ.'YES') GO TO 200                                     0009
      IF(MCAN.EQ.1) WRITE(NTP,1001)                                     0010
 1001 FORMAT(/T2,,DO YOU WANT TO USE (1) RASCHIG RINGS OR (2) BERL SADDL 0011
     1ES?',/T2,'ENTER 1 OR 2')                                          0012
      READ(NCR,1) NTYPE                                                 0013
      IF(NTYPE.GT.1) GO TO 101                                          0014
      IF(MCAN.EQ.1) WRITE(NTP,1002)                                     0015
 1002 FORMAT(/T2,'THE AVAILABLE RING SIZES ARE:',/T4,'(1) 12.7,(2) 25.4, 0016
     1(3) 38.1,(4) 50.8 (MM)',/T2,'ENTER APPROPRIATE CODE NO.')         0017
      READ(NCR,1) NPACK                                                 0018
      GO TO 999                                                         0019
  101 CONTINUE                                                          0020
      IF(MCAN.EQ.1) WRITE(NTP,1003)                                     0021
 1003 FORMAT(/T2,'THE AVAILABLE SADDLE SIZES ARE:',/T4,'(1) 12.7,(2) 25. 0022
     14,(3) 38.1 (MM)',/T2,'ENTER APPROPRIATE CODE NO.')                0023
      READ(NCR,1) NPACK                                                 0024
      GO TO 999                                                         0025

                                                      SUBROUTINE PACTYP
```

```
  200 CONTINUE                                                          0026
      IF(MCAN.EQ.1) WRITE(NTP,2001)                                     0027
 2001 FORMAT(/T2,'DO YOU HAVE THE PACKING DATA PREPARED?')              0028
      READ(NCR,2) PNAME                                                 0029
      IF(PNAME.EQ.'YES') GO TO 201                                      0030
      IF(MCAN.EQ.1) WRITE(NTP,2002)                                     0031
 2002 FORMAT(/T2,'REFER TO THE DISTILLATION PACKAGE MANUAL,SECTION 3.5,F 0032
     1OR FURTHER DETAILS.')                                             0033
      MCAN=100                                                          0034
      GO TO 999                                                         0035
  201 CONTINUE                                                          0036
      IF(MCAN.EQ.1) WRITE(NTP,2003)                                     0037
 2003 FORMAT(/T2,'ENTER CODES FOR PACKING TYPE AND SIZE')               0038
      READ(NCR,1) NTYPE,NPACK                                           0039
  999 CONTINUE                                                          0040
      RETURN                                                            0041
      END                                                               0042
```

SUBROUTINE PACTYP

```
      SUBROUTINE PAKING(NT,NP,CF,E,AMF1,AMF2,ANF1A,ANF1B,ANF2A,ANF2B,PF1     0001
     1,PF2,DS,PHTW,PHSW,SWP,BETA)                                           0002
C     ROUTINE SUPPLIES DATA ON EITHER CERAMIC RASCHIG RING OR BERL SADDLE
C     PACKING MATERIAL TO THE MAINLINE 'PACKED'.
C     CHOICE IS DEPENDENT ON TYPE (NT) AND SIZE (NP) OF PACKING SPECIFIED
C     IN MAINLINE PROGRAM.
C       REFERENCE: TREYBAL,R.E.,'MASS TRANSFER OPERATIONS',2ND ED.,
C                  MCGRAW-HILL,TABLES 6.2,6.3 AND 6.4.
      GO TO (100,200),NT                                                    0003
  100 CONTINUE                                                              0004
C     CERAMIC RASCHIG RINGS.
      GO TO (110,120,130,140),NP                                            0005
  110 CONTINUE                                                              0006
C     12.7 MM RINGS (0.5 IN)
      CF=640.                                                               0007
      E=0.63                                                                0008
      AMF1=8200.                                                            0009
      AMF2=9.32                                                             0010
      ANF1A=3.15                                                            0011
      ANF1B=-0.3                                                            0012
      ANF2A=0.151                                                           0013
      ANF2B=0.148                                                           0014
      PF1=-1.04                                                             0015
      PF2=-0.111                                                            0016
      DS=0.0582                                                             0017
      PHTW=2.25                                                             0018
      PHSW=0.00104                                                          0019
      SWP=1.21                                                              0020
      BETA=0.965*(DS**0.376)                                                0021
      GO TO 999                                                             0022
```

```
  120 CONTINUE                                                          0023
C     25.4 MM RINGS (1.0 IN)
      CF=160.                                                           0024
      E=0.73                                                            0025
      AMF1=0.274                                                        0026
      AMF2=463.                                                         0027
      ANF1A=0.0                                                         0028
      ANF1B=0.0                                                         0029
      ANF2A=0.528                                                       0030
      ANF2B=-0.0793                                                     0031
      PF1=0.552                                                         0032
      PF2=-0.47                                                         0033
      DS=0.1167                                                         0034
      PHTW=2.25                                                         0035
      PHSW=0.00104                                                      0036
      SWP=1.21                                                          0037
      BETA=0.965*(DS**0.376)                                            0038
      GO TO 999                                                         0039
```

```
  130 CONTINUE                                                          0040
C     38.1 MM RINGS (1.5 IN)
      CF=95.                                                            0041
      E=0.71                                                            0042
      ANF1=1.82                                                         0043
      ANF2=4.85                                                         0044
      ANF1A=0.675                                                       0045
      ANF1B=-0.1013                                                     0046
      ANF2A=0.148                                                       0047
      ANF2B=-0.022                                                      0048
      PF1=0.274                                                         0049
      PF2=0.140                                                         0050
      DS=0.1740                                                         0051
      PHTW=2.25                                                         0052
      PHSW=0.00104                                                      0053
      SWP=1.21                                                          0054
      BETA=0.965*(DS**0.376)                                            0055
      GO TO 999                                                         0056
```

```
  140 CONTINUE                                                          0057
C     50.8 MM RINGS (2.0 IN)
      CF=65.                                                            0058
      E=0.74                                                            0059
      AMF1=0.401                                                        0060
      AMF2=0.95                                                         0061
      ANF1A=0.0                                                         0062
      ANF1B=0.0                                                         0063
      ANF2A=0.0                                                         0064
      ANF2B=0.0                                                         0065
      PF1=0.481                                                         0066
      PF2=0.362                                                         0067
      DS=0.238                                                          0068
      PHTW=2.25                                                         0069
      PHSW=0.00104                                                      0070
      SWP=1.21                                                          0071
      BETA=0.965*(DS**0.376)                                            0072
      GO TO 999                                                         0073

                                                      SUBROUTINE PAKING
```

```
  200 CONTINUE                                                          0074
C     CERAMIC BERL SADDLES.
      GO TO (210,220,230),NP                                            0075
  210 CONTINUE                                                          0076
C     12.7 MM SADDLES (0.5 IN)
      CF=380.                                                           0077
      E=0.63                                                            0078
      AMF1=0.0336                                                       0079
      AMF2=2.54                                                         0080
      ANF1A=0.0                                                         0081
      ANF1B=0.0529                                                      0082
      ANF2A=0.0                                                         0083
      ANF2B=0.0529                                                      0084
      PF1=0.761                                                         0085
      PF2=0.170                                                         0086
      DS=0.0532                                                         0087
      PHTW=2.5                                                          0088
      PHSW=0.00032                                                      0089
      SWP=1.56                                                          0090
      BETA=0.965*(DS**0.376)                                            0091
      GO TO 999                                                         0092
```

```
  220 CONTINUE                                                          0093
C     25.4 MM SADDLES (1.0 IN)
      CF=110.                                                           0094
      E=0.69                                                            0095
      AMF1=15.89                                                        0096
      AMF2=238.                                                         0097
      ANF1A=0.686                                                       0098
      ANF1B=-0.1029                                                     0099
      ANF2A=0.420                                                       0100
      ANF2B=-0.063                                                      0101
      PF1=0.0                                                           0102
      PF2=-0.359                                                        0103
      DS=0.105                                                          0104
      PHTW=2.5                                                          0105
      PHSW=0.00032                                                      0106
      SWP=1.56                                                          0107
      BETA=0.965*(DS**0.376)                                            0108
      GO TO 999                                                         0109
```

```
  230 CONTINUE                                                          0110
C     38.1 MM SADDLES (1.5 IN)
      CF=65.                                                            0111
      E=0.75                                                            0112
      AMF1=0.613                                                        0113
      AMF2=465.                                                         0114
      ANF1A=0.0                                                         0115
      ANF1B=-0.0508                                                     0116
      ANF2A=0.325                                                       0117
      ANF2B=-0.0996                                                     0118
      PF1=0.455                                                         0119
      PF2=-0.1355                                                       0120
      DS=0.155                                                          0121
      PHTW=2.5                                                          0122
      PHSW=0.00032                                                      0123
      SWP=1.56                                                          0124
      BETA=0.965*(DS**0.376)                                            0125
      GO TO 999                                                         0126
  999 CONTINUE                                                          0127
      RETURN                                                            0128
      END                                                               0129
```

```
      SUBROUTINE FDATA(NCURVE,ABS,ORD)                                  0001
C     ROUTINE FITS AN N'TH ORDER POLYNOMIAL TO THE U.S. STONEWARE
C     COMPANY'S FLOODING AND LOADING CURVES FOR PACKED TOWERS.
C       REFERENCE: TREYBAL,R.E.,'MASS TRANSFER OPERATIONS',2ND ED.,
C                  MCGRAW-HILL,PG. 160,FIG. 6.26.
C     THE DESIRED VAPOUR PRESSURE DROP AT LOADING IS SPECIFIED IN
C     THE MAINLINE,AND COUNTER NCURVE SELECTS THE RELEVANT CURVE.
C     THE FIT IS BY LEAST SQUARES. POLYNOMIALS UP TO ORDER TEN ARE
C     TESTED,AND THE BEST FIT IS SELECTED.
C     THE ROUTINE REQUIRES UNIVAC 'MATH-PAC' LIBRARY SUBROUTINE 'GJR'
C     FOR SOLVING THE N'TH ORDER MATRICES INVOLVED IN THE LEAST SQUARES
C     PROCEDURE.
C     THIS OR A SIMILAR ROUTINE MUST BE PROVIDED.
      DIMENSION AMAT(20,20),CMAT(20),COEFF(20),X(50),Y(50)              0002
      DIMENSION JC(20),V(2)                                             0003
      DATA NCR,NTP/8,5/                                                 0004
    1 FORMAT( )                                                         0005
C       ADD DATA FOR CORRELATION CURVE SPECIFIED BY COUNTER NCURVE.
      GO TO (100,200,300,400,500,600),NCURVE                            0006
  100 CONTINUE                                                          0007
      CALL ERTRAN(6,'@ADD,E DISTFILE.FLOOD1 . ')                        0008
      GO TO 700                                                         0009
  200 CONTINUE                                                          0010
      CALL ERTRAN(6,'@ADD,E DISTFILE.FLOOD2 . ')                        0011
      GO TO 700                                                         0012
  300 CONTINUE                                                          0013
      CALL ERTRAN(6,'@ADD,E DISTFILE.FLOOD3 . ')                        0014
      GO TO 700                                                         0015
  400 CONTINUE                                                          0016
      CALL ERTRAN(6,'@ADD,E DISTFILE.FLOOD4 . ')                        0017
      GO TO 700                                                         0018
  500 CONTINUE                                                          0019
      CALL ERTRAN(6,'@ADD,E DISTFILE.FLOOD5 . ')                        0020
      GO TO 700                                                         0021
  600 CONTINUE                                                          0022
      CALL ERTRAN(6,'@ADD,E DISTFILE.FLOOD6 . ')                        0023
  700 CONTINUE                                                          0024
```

```
C     READ IN DATA.
      READ(NCR,1) NDATA                                                    0025
      READ(NCR,1) (X(I),Y(I),I=1,NDATA)                                    0026
      READ(NCR,1,END=701) IDUM                                             0027
  701 CONTINUE                                                             0028
C     SET INITIAL VALUES.
      NUM=2                                                                0029
      ERROR=1000.                                                          0030
      NTOT=20                                                              0031
  101 NUM1=NUM+1                                                           0032
C     'DO' LOOPS 10,11 & 12 CALCULATE TERMS IN MATRICES 'AMAT' AND 'CMAT'.
      DO 10 K=1,NUM                                                        0033
      DO 11 J=1,NUM                                                        0034
      SUMA=0.0                                                             0035
      SUMC=0.0                                                             0036
      DO 12 I=1,NDATA                                                      0037
      IF(J.EQ.1.AND.K.EQ.1) GO TO 102                                      0038
      SUMA=SUMA+X(I)**(J+K-2)                                              0039
      GO TO 103                                                            0040
  102 SUMA=NDATA                                                           0041
  103 IF(K.EQ.1) GO TO 104                                                 0042
      SUMC=SUMC+Y(I)*(X(I)**(K-1))                                         0043
      GO TO 12                                                             0044
  104 SUMC=SUMC+Y(I)                                                       0045
   12 CONTINUE                                                             0046
      AMAT(J,K)=SUMA                                                       0047
   11 CONTINUE                                                             0048
      CMAT(K)=SUMC                                                         0049
   10 CONTINUE                                                             0050

                                                          SUBROUTINE FDATA
```

```
C       INSERTION OF CMAT IN COLUMN NUM1 OF AMAT.
        DO 13 I=1,NUM                                                   0051
        AMAT(I,NUM1)=CMAT(I)                                            0052
   13   CONTINUE                                                        0053
C       'GJR' SOLVES EQUATIONS IN 'AMAT' BY GAUSS-JORDAN REDUCTION.
        V(1)=4                                                          0054
        CALL GJR(AMAT,NTOT,NTOT,NUM,NUM1,$900,JC,V)                     0055
        SUMERR=0.0                                                      0056
C       CALCULATE SUM OF SQUARED ERRORS.
        DO 20 J=1,NDATA                                                 0057
        SUMX=AMAT(1,NUM1)                                               0058
        DO 21 I=2,NUM                                                   0059
        SUMX=SUMX+AMAT(I,NUM1)*X(J)**(I-1)                              0060
   21   CONTINUE                                                        0061
        ERR=SUMX-Y(J)                                                   0062
        SQERR=ERR**2.                                                   0063
        SUMERR=SUMERR+SQERR                                             0064
   20   CONTINUE                                                        0065
C       COMPARE SUM OF SQUARED ERRORS WITH PREVIOUS VALUE.
        IF(SUMERR.LT.ERROR) GO TO 110                                   0066
  120   NUM=NUM+1                                                       0067
        IF(NUM.GT.11) GO TO 201                                         0068
        GO TO 101                                                       0069
  110   CONTINUE                                                        0070
C       STORE ERROR AND COEFFICIENTS OF POLYNOMIAL WITH BEST FIT.
        ERROR=SUMERR                                                    0071
        NPOLY=NUM                                                       0072
        DO 30 I=1,NUM                                                   0073
        COEFF(I)=AMAT(I,NUM1)                                           0074
   30   CONTINUE                                                        0075
        GO TO 120                                                       0076
  201   CONTINUE                                                        0077
```

```
C       CALCULATION OF ORDINATE CORRESPONDING TO 'ABS' ON CURVE 'NCURVE'.
        NORD=NPOLY-1                                                    0078
        XSUM=COEFF(1)                                                   0079
        DO 40 I=2,NPOLY                                                 0080
        IC=I-1                                                          0081
        XFACT=COEFF(I)*(ABS**IC)                                        0082
        XSUM=XSUM+XFACT                                                 0083
   40   CONTINUE                                                        0084
        ORD=XSUM                                                        0085
        WRITE(NTP,1001) NORD,ERROR                                      0086
 1001   FORMAT(/T2,'POLYNOMIAL OF ORDER',I3,' FITTED.',/T2,            0087
       1'SUM OF SQUARED ERRORS=',F15.10)                                0088
        GO TO 999                                                       0089
C       ERROR MESSAGES ************************************************
C       ERROR DETECTED IN 'GJR',
  900   CONTINUE                                                        0090
        WRITE(NTP,9000) JC(1)                                           0091
 9000   FORMAT(/T2,'ERROR IN MATRIX,,,JC=',I3)                          0092
  999   CONTINUE                                                        0093
        RETURN                                                          0094
        END                                                             0095
```

```
      SUBROUTINE RHOLIQ(NPHY,NR,NL,NH,TG,X,RHO1,RHO2,RHOM)                0001
C     ROUTINE CALCULATES LIQUID DENSITIES OF SINGLE COMPONENTS OR BINARY
C     SOLUTIONS OF COMPOSITION X AT TEMPERATURE TG.
C     KNOWN DATA CAN BE SUPPLIED IN SEVERAL WAYS.
C     FOR DETAILS REFER TO THE PACKAGE MANUAL,SECTION 3.
      DIMENSION AMW(2),EXP1(4),RHOL1(50),RHOL2(50),RHOS(50)              0002
      DIMENSION TC(2),TRHO1(50),TRHO2(50),XS(50)                         0003
    1 FORMAT( )                                                          0004
      NCR=8                                                              0005
      EXP1(1)=0.25                                                       0006
      EXP1(2)=0.29                                                       0007
      EXP1(3)=0.31                                                       0008
      EXP1(4)=0.33                                                       0009
      IF(NPHY.GT.1) GO TO 600                                            0010
C     DATA REQUIRED FOR EACH COMPONENT:
C       (A) CRITICAL TEMPERATURE (DEG C)
C       (B) NO. OF TEMPERATURE-DENSITY DATA PAIRS TO BE SUPPLIED.
      CALL ERTRAN(6,'@ADD,E DISTFILE,ROLDA1 . ')                         0011
      READ(NCR,1) (AMW(I),TC(I),I=1,2)                                   0012
      TC(1)=TC(1)+273.                                                   0013
      TC(2)=TC(2)+273.                                                   0014
      READ(NCR,1) NDAT1,NDAT2                                            0015
      READ(NCR,1,END=600) IDUM                                           0016
  600 CONTINUE                                                           0017
      T=TG+273.                                                          0018
      IF(NDAT1.GT.1) GO TO 100                                           0019
      IF(NPHY.GT.1) GO TO 611                                            0020
C     LIGHT COMPONENT DENSITY. (ONLY ONE DATA PAIR SUPPLIED)
      CALL ERTRAN(6,'@ADD,E DISTFILE,ROLDA2 . ')                         0021
      READ(NCR,1) TRHO1(1),RHOL1(1)                                      0022
      TRHO1(1)=TRHO1(1)+273.                                             0023
  611 CONTINUE                                                           0024
      RHO1=RHOL1(1)*((TC(1)-T)/(TC(1)-TRHO1(1)))**EXP1(NL)               0025
      GO TO 199                                                          0026
```

```
  100 CONTINUE                                                          0027
      IF(NPHY.GT.1) GO TO 612                                           0028
C     LIGHT COMPONENT DENSITY, (MORE THAN ONE DATA PAIR SUPPLIED)
      CALL ERTRAN(6,'@ADD E DISTFILE,ROLDA2 , ')                        0029
      READ(NCR,1) (TRHO1(I),RHOL1(I),I=1,NDAT1)                         0030
  612 CONTINUE                                                          0031
      IF(TG.GT.TRHO1(1)) GO TO 110                                      0032
      RHO1=RHOL1(1)                                                     0033
      GO TO 199                                                         0034
  110 CONTINUE                                                          0035
      DO 10 I=1,NDAT1                                                   0036
      DIFF=TG-TRHO1(I)                                                  0037
      IF(DIFF.LE.0.0) GO TO 111                                         0038
   10 CONTINUE                                                          0039
      GO TO 120                                                         0040
  111 CONTINUE                                                          0041
      TFACT=(TRHO1(I)-TG)/(TRHO1(I)-TRHO1(I-1))                         0042
      RHO1=RHOL1(I)+(RHOL1(I-1)-RHOL1(I))*TFACT                         0043
      GO TO 199                                                         0044
  120 CONTINUE                                                          0045
      RHO1=RHOL1(NDAT1)                                                 0046
  199 CONTINUE                                                          0047
      IF(NDAT2.GT.1) GO TO 200                                          0048
      IF(NPHY.GT.1) GO TO 601                                           0049
C     HEAVY COMPONENT DENSITY, (ONLY ONE DATA PAIR SUPPLIED)
      READ(NCR,1) TRHO2(1),RHOL2(1)                                     0050
      TRHO2(1)=TRHO2(1)+273.                                            0051
      READ(NCR,1,END=601) IDUM                                          0052
  601 CONTINUE                                                          0053
      RHO2=RHOL2(1)*((TC(2)-T)/(TC(2)-TRHO2(1)))**EXP1(NH)              0054
      GO TO 299                                                         0055
```

SUBROUTINE RHOLIQ

```
  200 CONTINUE                                                          0056
      IF(NPHY.GT.1) GO TO 602                                           0057
C     HEAVY COMPONENT DENSITY. (MORE THAN ONE DATA PAIR SUPPLIED)
      READ(NCR,1) (TRHO2(I),RHOL2(I),I=1,NDAT2)                         0058
      READ(NCR,1,END=602) IDUM                                          0059
  602 CONTINUE                                                          0060
      IF(TG.GT.TRHO2(1)) GO TO 210                                      0061
      RHO2=RHOL2(1)                                                     0062
      GO TO 299                                                         0063
  210 CONTINUE                                                          0064
      DO 20 I=1,NDAT2                                                   0065
      DIFF=TG-TRHO2(I)                                                  0066
      IF(DIFF.LE.0.0) GO TO 211                                         0067
   20 CONTINUE                                                          0068
      GO TO 220                                                         0069
  211 CONTINUE                                                          0070
      TFACT=(TRHO2(I)-TG)/(TRHO2(I)-TRHO2(I-1))                         0071
      RHO2=RHOL2(I)+(RHOL2(I-1)-RHOL2(I))*TFACT                         0072
      GO TO 299                                                         0073
  220 CONTINUE                                                          0074
      RHO2=RHOL2(NDAT2)                                                 0075
  299 CONTINUE                                                          0076
      IF(NR.EQ.1) GO TO 300                                             0077
C     DENSITY OF SOLUTION OF COMPOSITION X AT TEMPERATURE TG.
      FACT=X*AMW(1)*RHO2+(1.-X)*AMW(2)*RHO1                             0078
      RHOM=RHO1*RHO2*(X*AMW(1)+(1.-X)*AMW(2))/FACT                      0079
      GO TO 999                                                         0080
```

```
  300 CONTINUE                                                          0081
C     OPTIONAL FACILITY FOR SYSTEMS WITH COMPOSITION-DENSITY DATA KNOWN
C     AT SOME TEMPERATURE 'TBASEC'.
      IF(NPHY.GT.1) GO TO 603                                           0082
C     'NDATA' COMPOSITION-DENSITY DATA PAIRS,AT TEMPERATURE TBASEC
C     MUST BE SUPPLIED.
      CALL ERTRAN(6,'@ADD,E DISTFILE,ROLDA3 . ')                        0083
      READ(NCR,1) NDATA,TBASEC                                          0084
      READ(NCR,1) (XS(I),RHOS(I),I=1,NDATA)                             0085
      READ(NCR,1,END=603) IDUM                                          0086
  603 CONTINUE                                                          0087
      TBASE=TBASEC+273.                                                 0088
      IF(X.GT.XS(1)) GO TO 310                                          0089
      RHOC=RHOS(1)                                                      0090
      GO TO 399                                                         0091
  310 CONTINUE                                                          0092
      DO 30 I=1,NDATA                                                   0093
      DIFF=X-XS(I)                                                      0094
      IF(DIFF.LE.0.0) GO TO 311                                         0095
   30 CONTINUE                                                          0096
      GO TO 320                                                         0097
  311 CONTINUE                                                          0098
      XFACT=(XS(I)-X)/(XS(I)-XS(I-1))                                   0099
      RHOC=RHOS(I)+(RHOS(I-1)-RHOS(I))*XFACT                            0100
      GO TO 399                                                         0101
  320 CONTINUE                                                          0102
      RHOC=RHOS(NDATA)                                                  0103
  399 CONTINUE                                                          0104
C     DENSITY OF SOLUTION OF COMPOSITION X AT TEMPERATURE TG.
      TCP=X*TC(1)+(1.-X)*TC(2)                                          0105
      RHOM=RHOC*((TCP-T)/(TCP-TBASE))**((EXP1(NL)+EXP1(NH))/2.)         0106
  999 CONTINUE                                                          0107
      RETURN                                                            0108
      END                                                               0109
```

SUBROUTINE RHOLIQ

```
      SUBROUTINE SURTEN(NPHY,NL,NH,TG,X,SIGM)                          0001
C     ROUTINE CALCULATES THE SURFACE TENSION OF BINARY SOLUTIONS OF
C     COMPOSITION X AT TEMPERATURE TG.
C     KNOWN DATA CAN BE SUPPLIED IN SEVERAL WAYS.
C     FOR DETAILS REFER TO THE PACKAGE MANUAL,SECTION 3.
      DIMENSION AMW(2),PARA(2),SIG1(50),SIG2(50),SIGS(50)              0002
      DIMENSION T1(2),T2(2),TC(50),XS(50)                              0003
    1 FORMAT( )                                                        0004
      NCR=8                                                            0005
      IF(NPHY.GT.1) GO TO 600                                          0006
      CALL ERTRAN(6,'@ADD E DISTFILE,SIGDA1 , ')                       0007
      READ(NCR,1) NSIGM                                                0008
      READ(NCR,1) (AMW(I),TC(I),PARA(I),I=1,2)                         0009
      READ(NCR,1) NDAT1,NDAT2                                          0010
      TC(1)=TC(1)+273.                                                 0011
      TC(2)=TC(2)+273.                                                 0012
      READ(NCR,1,END=600) IDUM                                         0013
  600 CONTINUE                                                         0014
      T=TG+273.                                                        0015
      IF(NSIGM.EQ.2) GO TO 300                                         0016
      IF(PARA(1).GT.0.0) GO TO 198                                     0017
      IF(NDAT1.GT.1) GO TO 100                                         0018
      IF(NPHY.GT.1) GO TO 601                                          0019
      CALL ERTRAN(6,'@ADD E DISTFILE,SIGDA2 , ')                       0020
      READ(NCR,1) T1(1),SIG1(1)                                        0021
      T1(1)=T1(1)+273.                                                 0022
      READ(NCR,1,END=601) IDUM                                         0023
  601 CONTINUE                                                         0024
      SIGA=SIG1(1)*((TC(1)-T)/(TC(1)-T1(1)))**1.2                      0025
      GO TO 199                                                        0026
```

```
100 CONTINUE                                                          0027
    IF(NPHY.GT.1) GO TO 602                                           0028
    CALL ERTRAN(6,'@ADD E DISTFILE.SIGDA2 . ')                        0029
    READ(NCR,1) (T1(I),SIG1(I),I=1,NDAT1)                             0030
    READ(NCR,1,END=602) IDUM                                          0031
602 CONTINUE                                                          0032
    IF(TG.GT.T1(1)) GO TO 110                                         0033
    SIGA=SIG1(1)                                                      0034
    GO TO 199                                                         0035
110 CONTINUE                                                          0036
    DO 10 I=1,NDAT1                                                   0037
    DIFF=TG-T1(I)                                                     0038
    IF(DIFF.LE.0.0) GO TO 111                                         0039
 10 CONTINUE                                                          0040
    GO TO 120                                                         0041
111 CONTINUE                                                          0042
    TFACT=(T1(I)-TG)/(T1(I)-T1(I-1))                                  0043
    SIGA=SIG1(I)+(SIG1(I-1)-SIG1(I))*TFACT                            0044
    GO TO 199                                                         0045
120 CONTINUE                                                          0046
    SIGA=SIG1(NDAT1)                                                  0047
    GO TO 199                                                         0048
198 CONTINUE                                                          0049
    NR=0                                                              0050
    CALL RHOLIQ(NPHY,NR,NL,NH,TG,X,RHO1,RHO2,RHOM)                    0051
    SIGA=(PARA(1)*RHO1/AMW(1))**4.                                    0052
199 CONTINUE                                                          0053
    IF(PARA(2).GT.0.0) GO TO 400                                      0054
    IF(NDAT2.GT.1) GO TO 200                                          0055
    IF(NPHY.GT.1) GO TO 604                                           0056
    CALL ERTRAN(6,'@ADD E DISTFILE.SIGDA4 . ')                        0057
    READ(NCR,1) T2(1),SIG2(1)                                         0058
    T2(1)=T2(1)+273.                                                  0059
    READ(NCR,1,END=604) IDUM                                          0060
604 CONTINUE                                                          0061
    SIGB=SIG2(1)*((TC(2)-T)/(TC(2)-T2(1)))**1.2                       0062
    GO TO 299                                                         0063
```

```
  200 CONTINUE                                                          0064
      IF(NPHY.GT.1) GO TO 605                                           0065
      CALL ERTRAN(6,'@ADD E DISTFILE,SIGDA4 , ')                        0066
      READ(NCR,1) (T2(I),SIG2(I),I=1,NDAT2)                             0067
      READ(NCR,1,END=605) IDUM                                          0068
  605 CONTINUE                                                          0069
      IF(TG.GT.T2(1)) GO TO 210                                         0070
      SIGB=SIG2(1)                                                      0071
      GO TO 299                                                         0072
  210 CONTINUE                                                          0073
      DO 20 I=1,NDAT2                                                   0074
      DIFF=TG-T2(I)                                                     0075
      IF(DIFF.LE.0.0) GO TO 211                                         0076
   20 CONTINUE                                                          0077
      GO TO 220                                                         0078
  211 CONTINUE                                                          0079
      TFACT=(T2(I)-TG)/(T2(I)-T2(I-1))                                  0080
      SIGB=SIG2(I)+(SIG2(I-1)-SIG2(I))*TFACT                            0081
      GO TO 299                                                         0082
  220 CONTINUE                                                          0083
      SIGB=SIG2(NDAT2)                                                  0084
  299 CONTINUE                                                          0085
      SIGN=SIGA*SIGB/(X*SIGB+(1.-X)*SIGA)                               0086
      GO TO 999                                                         0087
  300 CONTINUE                                                          0088
C     OPTIONAL FACILITY FOR SYSTEMS WITH COMPOSITION-SURFACE TENSION
C     DATA KNOWN AT SOME TEMPERATURE 'TBASEC'.
      IF(NPHY.GT.1) GO TO 603                                           0089
C     'NDATA' COMPOSITION-SURFACE TENSION DATA PAIRS,AT TEMPERATURE
C     TBASEC,MUST BE SUPPLIED.
```

```
      CALL ERTRAN(6,'@ADD,E DISTFILE,SIGDA3 . ')                        0090
      READ(NCR,1) NDATA,TBASEC                                          0091
      READ(NCR,1) (XS(I),SIGS(I),I=1,NDATA)                             0092
      READ(NCR,1,END=603) IDUM                                          0093
  603 CONTINUE                                                          0094
      TBASE=TBASEC+273.                                                 0095
      IF(X.GT.XS(1)) GO TO 310                                          0096
      SIGC=SIGS(1)                                                      0097
      GO TO 399                                                         0098
  310 CONTINUE                                                          0099
      DO 30 I=1,NDATA                                                   0100
      DIFF=X-XS(I)                                                      0101
      IF(DIFF.LE.0.0) GO TO 311                                         0102
   30 CONTINUE                                                          0103
      GO TO 320                                                         0104
  311 CONTINUE                                                          0105
      XFACT=(XS(I)-X)/(XS(I)-XS(I-1))                                   0106
      SIGC=SIGS(I)+(SIGS(I-1)-SIGS(I))*XFACT                            0107
      GO TO 399                                                         0108
  320 CONTINUE                                                          0109
      SIGC=SIGS(NDATA)                                                  0110
  399 CONTINUE                                                          0111
      TCP=X*TC(1)+(1.-X)*TC(2)                                          0112
      SIGM=SIGC*((TCP-T)/(TCP-TBASE))**1.2                              0113
      GO TO 999                                                         0114
  400 CONTINUE                                                          0115
      NR=0                                                              0116
      CALL RHOLIQ(NPHY,NR,NL,NH,TG,X,RHO1,RHO2,RHOM)                    0117
      SIGB=(PARA(2)*RHO2/AMW(2))**4.                                    0118
      AMWL=X*AMW(1)+(1.-X)*AMW(2)                                       0119
      PARSUM=X*PARA(1)+(1.-X)*PARA(2)                                   0120
      SIGM=(PARSUM*RHOM/AMWL)**4.                                       0121
  999 CONTINUE                                                          0122
      RETURN                                                            0123
      END                                                               0124
```

```
      SUBROUTINE DIFLIQ(NPHY,NL,NH,TG,X,DL)                              0001
C     ROUTINE CALCULATES LIQUID DIFFUSIVITY OF BINARY SOLUTIONS
C     OF COMPOSITION X AT TEMPERATURE TG.
C     FOR DATA REQUIRED REFER TO PACKAGE MANUAL,SECTION 3.
    1 FORMAT( )                                                          0002
      NCR=8                                                              0003
      IF(NPHY.GT.1) GO TO 600                                            0004
      CALL ERTRAN(6,'@ADD,E DISTFILE,DLDAT . ')                          0005
      READ(NCR,1) DL12,TDL                                               0006
      TDL=TDL+273.                                                       0007
      READ(NCR,1,END=600) IDUM                                           0008
  600 CONTINUE                                                           0009
      TAG=TDL-273.                                                       0010
      T=TG+273.                                                          0011
      TA=TDL                                                             0012
      CALL VISCL(NPHY,NL,NH,TAG,X,AMU1,AMU2,AMUM)                        0013
      AMUA=AMU2                                                          0014
      DL=DL12/(10.**5)                                                   0015
      CONST=DL*AMUA/TA                                                   0016
      NPH=2                                                              0017
      CALL VISCL(NPH,NL,NH,TG,X,AMU1,AMU2,AMUM)                          0018
      DL=CONST*T/AMUM                                                    0019
  999 CONTINUE                                                           0020
      RETURN                                                             0021
      END                                                                0022
```

```
      SUBROUTINE DIFVAP(NPHY,PT,TG,DV)                                  0001
C     ROUTINE CALCULATES VAPOUR DIFFUSIVITY OF BINARY SOLUTIONS
C     OF COMPOSITION X AT TEMPERATURE TG AND TOTAL PRESSURE PT.
C     FOR DATA REQUIRED REFER TO PACKAGE MANUAL,SECTION 3.
      DIMENSION AMW(2),TB(2),VB(2)                                      0002
    1 FORMAT( )                                                         0003
      NCR=8                                                             0004
      IF(NPHY.GT.1) GO TO 600                                           0005
      CALL ERTRAN(6,'@ADD,E DISTFILE,DVDAT . ')                         0006
      READ(NCR,1) (AMW(I),VB(I),TB(I),I=1,2)                            0007
      DO 10 I=1,2                                                       0008
      VB(I)=VB(I)**0.333                                                0009
      TB(I)=TB(I)+273.                                                  0010
   10 CONTINUE                                                          0011
      READ(NCR,1,END=600) IDUM                                          0012
  600 CONTINUE                                                          0013
      T=TG+273.                                                         0014
      P=PT/760.                                                         0015
      B=((AMW(1)+AMW(2))/(AMW(1)*AMW(2)))**0.5                          0016
      F=((2.*(VB(1)*VB(2))**0.5)/(VB(1)+VB(2)))**3.                     0017
      C12=1.47*F*((TB(1)*TB(2))**0.5)                                   0018
      DV=((0.00837*B)/(P*((VB(1)+VB(2))**2.)))*((T**2.5)/(T+C12))       0019
      RETURN                                                            0020
      END                                                               0021
```

SUBROUTINE DIFVAP

```
      SUBROUTINE VISCL(NPHY,NL,NH,TG,X,AMUA,AMUB,AMUM)                  0001
C     ROUTINE CALCULATES LIQUID VISCOSITIES OF SINGLE COMPONENTS OR
C     BINARY SOLUTIONS OF COMPOSITION X AT TEMPERATURE TG.
C     FOR DETAILS REFER TO THE PACKAGE MANUAL,SECTION 3.
      DIMENSION AMU1(50),AMU2(50),B(2),RHO(2)                           0002
      DIMENSION T1(50),T2(50),TC(2),TR(2)                               0003
    1 FORMAT( )                                                         0004
      NCR=8                                                             0005
      T1G=TG                                                            0006
      T2G=TG                                                            0007
      IF(NPHY.GT.1) GO TO 611                                           0008
      CALL ERTRAN(6,'@ADD,E DISTFILE,FMUDA1 . ')                        0009
      READ(NCR,1) (TC(I),B(I),I=1,2)                                    0010
      READ(NCR,1) NDAT1,NDAT2                                           0011
      READ(NCR,1,END=601) IDUM                                          0012
  601 CONTINUE                                                          0013
      DO 11 I=1,2                                                       0014
      TC(I)=TC(I)+273.                                                  0015
   11 CONTINUE                                                          0016
  611 CONTINUE                                                          0017
      T=TG+273.                                                         0018
      DO 12 I=1,2                                                       0019
      TR(I)=T/TC(I)                                                     0020
   12 CONTINUE                                                          0021
      IF(B(1).GT.0.0) GO TO 198                                         0022
      IF(NPHY.GT.1) GO TO 602                                           0023
      CALL ERTRAN(6,'@ADD,E DISTFILE,FMUDA2 . ')                        0024
      READ(NCR,1) (T1(I),AMU1(I),I=1,NDAT1)                             0025
      READ(NCR,1,END=602) IDUM                                          0026
  602 CONTINUE                                                          0027
      IF(TG.GT.T1(1)) GO TO 110                                         0028
      AMUA=AMU1(1)                                                      0029
      GO TO 199                                                         0030
```

```
110 CONTINUE                                                          0031
    DO 10 I=1,NDAT1                                                   0032
    DIFF=TG-T1(I)                                                     0033
    IF(DIFF.LE.0.0) GO TO 111                                         0034
 10 CONTINUE                                                          0035
    GO TO 120                                                         0036
111 CONTINUE                                                          0037
    TFACT=(T1(I)-TG)/(T1(I)-T1(I-1))                                  0038
    AMUA=AMU1(I)+(AMU1(I-1)-AMU1(I))*TFACT                            0039
    GO TO 199                                                         0040
120 CONTINUE                                                          0041
    AMUA=AMU1(NDAT1)                                                  0042
    GO TO 199                                                         0043
198 CONTINUE                                                          0044
    NR=0                                                              0045
    CALL RHOLIQ(NPHY,NR,NL,NH,TIG,X,RHO1,RHO2,RHOM)                   0046
    RHO(1)=RHO1                                                       0047
    ALFA=B(1)*(1.-TR(1))/TR(1)                                        0048
    AMUA=(0.1167*(RHO(1)**0.5))*(10.**ALFA)                           0049
199 CONTINUE                                                          0050
    IF(B(2).GT.0.0) GO TO 300                                         0051
    IF(NPHY.GT.1) GO TO 603                                           0052
    CALL ERTRAN(6,'@ADD,E DISTFILE,FMUDA3 , ')                        0053
    READ(NCR,1) (T2(I),AMU2(I),I=1,NDAT2)                             0054
    READ(NCR,1,END=603) IDUM                                          0055
603 CONTINUE                                                          0056
    IF(TG.GT.T2(1)) GO TO 210                                         0057
    AMUB=AMU2(1)                                                      0058
    GO TO 299                                                         0059
```

SUBROUTINE VISCL

```
210 CONTINUE                                                          0060
    DO 20 I=1,NDAT2                                                   0061
    DIFF=TG-T2(I)                                                     0062
    IF(DIFF.LE.0.0) GO TO 211                                         0063
 20 CONTINUE                                                          0064
    GO TO 220                                                         0065
211 CONTINUE                                                          0066
    TFACT=(T2(I)-TG)/(T2(I)-T2(I-1))                                  0067
    AMUB=AMU2(I)+(AMU2(I-1)-AMU2(I))*TFACT                            0068
    GO TO 299                                                         0069
220 CONTINUE                                                          0070
    AMUB=AMU2(NDAT2)                                                  0071
299 CONTINUE                                                          0072
    AMUM=(X*(AMUA**(1./3.))+(1.-X)*(AMUB**(1./3.)))**3.               0073
    GO TO 999                                                         0074
300 CONTINUE                                                          0075
    NR=0                                                              0076
    CALL RHOLIQ(NPHY,NR,NL,NH,T2G,X,RHO1,RHO2,RHOM)                   0077
    RHO(2)=RHO2                                                       0078
    ALFB=B(2)*(1.-TR(2))/TR(2)                                        0079
    AMUB=(0.1167*(RHO(2)**0.5))*(10.**ALFB)                           0080
    AMUM=(X*(AMUA**(1./3.))+(1.-X)*(AMUB**(1./3.)))**3.               0081
999 CONTINUE                                                          0082
    RETURN                                                            0083
    END                                                               0084
```

SUBROUTINE VISCL

```
      SUBROUTINE VISCV(NPHY,TG,Y,AMUM)                                  0001
C     ROUTINE CALCULATES VAPOUR VISCOSITIES OF BINARY SOLUTIONS
C     OF COMPOSITION X AT TEMPERATURE TG.
C     FOR DATA REQUIRED REFER TO PACKAGE MANUAL,SECTION 3.
      DIMENSION AMU1(50),AMU2(50),AMUV(2),AMW(2)                        0002
      DIMENSION T1(50),T2(50),TB(2),VB(2),YI(2)                         0003
    1 FORMAT( )                                                         0004
      NCR=8                                                             0005
      YI(1)=Y                                                           0006
      YI(2)=1.-Y                                                        0007
      IF(NPHY.GT.1) GO TO 601                                           0008
      CALL ERTRAN(6,'@ADD,E DISTFILE.VMUDA1 . ')                        0009
      READ(NCR,1) (AMW(I),VB(I),TB(I),I=1,2)                            0010
      READ(NCR,1) NDAT1,NDAT2                                           0011
      TB(1)=TB(1)+273.                                                  0012
      TB(2)=TB(2)+273.                                                  0013
      READ(NCR,1,END=601) IDUM                                          0014
  601 CONTINUE                                                          0015
      T=TG+273.                                                         0016
      IF(VB(1).GT.0.0) GO TO 199                                        0017
      IF(NPHY.GT.1) GO TO 602                                           0018
      CALL ERTRAN(6,'@ADD,E DISTFILE.VMUDA2 . ')                        0019
      READ(NCR,1) (T1(I),AMU1(I),I=1,NDAT1)                             0020
      READ(NCR,1,END=602) IDUM                                          0021
  602 CONTINUE                                                          0022
      IF(TG.GT.T1(1)) GO TO 110                                         0023
      AMUV(1)=AMU1(1)                                                   0024
      GO TO 199                                                         0025
```

```
110 CONTINUE                                                          0026
    DO 10 I=1,NDAT1                                                   0027
    DIFF=TG-T1(I)                                                     0028
    IF(DIFF.LE.0.0) GO TO 111                                         0029
 10 CONTINUE                                                          0030
    GO TO 120                                                         0031
111 CONTINUE                                                          0032
    TFACT=(T1(I)-TG)/(T1(I)-T1(I-1))                                  0033
    AMUV(1)=AMU1(I)+(AMU1(I-1)-AMU1(I))*TFACT                         0034
    GO TO 199                                                         0035
120 CONTINUE                                                          0036
    AMUV(1)=AMU1(NDAT1)                                               0037
199 CONTINUE                                                          0038
    IF(VB(2).GT.0.0) GO TO 300                                        0039
    IF(NPHY.GT.1) GO TO 603                                           0040
    CALL ERTRAN(6,'@ADD,E DISTFILE,VMUDA3 . ')                        0041
    READ(NCR,1) (T2(I),AMU2(I),I=1,NDAT2)                             0042
    READ(NCR,1,END=603) IDUM                                          0043
603 CONTINUE                                                          0044
    IF(TG.GT.T2(1)) GO TO 210                                         0045
    AMUV(2)=AMU2(1)                                                   0046
    GO TO 999                                                         0047
210 CONTINUE                                                          0048
    DO 20 I=1,NDAT2                                                   0049
    DIFF=TG-T2(I)                                                     0050
    IF(DIFF.LE.0.0) GO TO 211                                         0051
 20 CONTINUE                                                          0052
    GO TO 220                                                         0053
```

```
211 CONTINUE                                                          0054
    TFACT=(T2(I)-TG)/(T2(I)-T2(I-1))                                  0055
    AMUV(2)=AMU2(I)+(AMU2(I-1)-AMU2(I))*TFACT                         0056
    GO TO 999                                                         0057
220 CONTINUE                                                          0058
    AMUV(2)=AMU2(NDAT2)                                               0059
    GO TO 999                                                         0060
300 CONTINUE                                                          0061
    DO 30 I=1,2                                                       0062
    FACT1=27.*(AMW(I)**0.5)*(T**1.5)                                  0063
    FACT2=(VB(I)**0.667)*(T+1.47*TB(I))                               0064
    AMUV(I)=FACT1/FACT2                                               0065
 30 CONTINUE                                                          0066
999 CONTINUE                                                          0067
    SUMA=0.0                                                          0068
    SUMB=0.0                                                          0069
    DO 90 I=1,2                                                       0070
    SUMA=SUMA+YI(I)*(AMW(I)**0.5)*AMUV(I)                             0071
    SUMB=SUMB+YI(I)*(AMW(I)**0.5)                                     0072
 90 CONTINUE                                                          0073
    AMUM=SUMA/SUMB                                                    0074
    RETURN                                                            0075
    END                                                               0076
```

SUBROUTINE VISCV

```
      SUBROUTINE RHOH2O(NPHY,T,RHO)                                      0001
C     ROUTINE CALCULATES DENSITY OF WATER AT TEMPERATURE T.
C     A SET OF NDATA TEMPERATURE-DENSITY DATA PAIRS MUST BE SUPPLIED.
      DIMENSION RHOW(200),TW(200)                                        0002
    1 FORMAT( )                                                          0003
      NCR=8                                                              0004
      IF(NPHY.GT.1) GO TO 600                                            0005
      CALL ERTRAN(6,'@ADD E DISTFILE,H2ORHO . ')                         0006
      READ(NCR,1) NDATA                                                  0007
      READ(NCR,1) (TW(I),RHOW(I),I=1,NDATA)                              0008
      READ(NCR,1,END=600) IDUM                                           0009
  600 CONTINUE                                                           0010
      IF(T.GT.TW(1)) GO TO 100                                           0011
      RHO=RHOW(1)                                                        0012
      GO TO 999                                                          0013
  100 CONTINUE                                                           0014
      DO 10 I=1,NDATA                                                    0015
      DIFF=T-TW(I)                                                       0016
      IF(DIFF.LE.0.0) GO TO 111                                          0017
   10 CONTINUE                                                           0018
      GO TO 120                                                          0019
  111 CONTINUE                                                           0020
      TFACT=(TW(I)-T)/(TW(I)-TW(I-1))                                    0021
      RHO=RHOW(I)+(RHOW(I-1)-RHOW(I))*TFACT                              0022
      GO TO 999                                                          0023
  120 CONTINUE                                                           0024
      RHO=RHOW(NDATA)                                                    0025
  999 CONTINUE                                                           0026
      RETURN                                                             0027
      END                                                                0028
```

SUBROUTINE RHOH2O

```
C     DIMENSION AND COMMON BLOCK 'MULTICOM' FOR MULTICOMPONENT DISTILLATION
C     PROGRAM,
      PARAMETER MC=30                                                          0001
      PARAMETER MP=53                                                          0002
      DIMENSION A(MC,MP),AN(MC),AK1(MC),AK2(MC),AK3(MC),AK4(MC)                0003
      DIMENSION B(MC),BD(MC),BN(MC),CN(MC),D(MC)                               0004
      DIMENSION EFFV(MC,MP),HLIQ(MP),HVAP(MP),K(MC,MP),KF(MC)                  0005
      DIMENSION HL1(MC),HL2(MC),HL3(MC),HV1(MC),HV2(MC),HV3(MC)                0006
      DIMENSION LB(MC,MP),LD(MC,MP),LMFR(MC,MP),LRTOT(MP),LSTOT(MP)            0007
      DIMENSION LPROF(MP),PRESS(MP),PRES(MP),S(MC,MP),SLIQ(MP),SVAP(MP)        0008
      DIMENSION TEMP(MP),TITLES(MC,5)                                          0009
      DIMENSION VB(MC,MP),VD(MC,MP),VMFR(MC,MP),VRTOT(MP),VSTOT(MP)           0010
      DIMENSION VPROF(MP),X(MC,MP),XF(MC),Y(MC,MP),YF(MC),Z(MC)               0011
      COMMON A,ACCLIM,AKB1,AKB2,AKB3,AKB4,AK1,AK2,AK3,AK4,ALIQ,AN,AVAP         0012
      COMMON B,BD,BM,BN,BTOT                                                   0013
      COMMON CN,D,DELTAC,DELTAP,DM,DTOT,EFFV,FENTH,FTOT                        0014
      COMMON HD,HFEED,HLIQ,HL1,HL2,HL3,HVAP,HV1,HV2,HV3                        0015
      COMMON ICOUNT,K,KF,LB,LD,LMFR,LPROF,LRTOT,LSTOT,MEFF                     0016
      COMMON NC,NCOND,NENTH,NEQUIL,NF,NF1,NF2,NJ,NOWN,NP,NP1,NP2               0017
      COMMON NQVAL,NUM1,NUM2,NUM3                                              0018
      COMMON NCR,NG1,NG2,NG3,NG4,NLP,NTP                                       0019
      COMMON PCOND,PRESS,PRES                                                  0020
      COMMON Q,QC,QF,QR                                                        0021
      COMMON RR,S,SLIQ,SVAP                                                    0022
      COMMON TBOT,TEMP,TFEED,THETA,TITLES,TTOP                                 0023
      COMMON VB,VD,VMFR,VPROF,VRTOT,VSTOT                                      0024
      COMMON X,XF,Y,YF,Z                                                       0025

                                                        COMMON 'MULTICOM'
```

```
C      MAINLINE PROGRAM FOR SOLUTION OF MULTICOMPONENT DISTILLATION
C      PROBLEMS BY THE THIELE-GEDDES METHOD.
C      THE HOLLAND 'THETA' METHOD IS USED TO ACCELERATE CONVERGENCE.
C      FOR FURTHER DETAILS REFER TO THE DISTILLATION-PACKAGE USER'S MANUAL.
C      DEPARTMENT OF CHEMICAL ENGINEERING,UNIVERSITY OF CAPE TOWN.
C      ***************************************************
C      INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C      ***************************************************
       DIMENSION COMPNO(MC),TRAYNO(MP)                                    0001
       REAL K,KF,LNFR,LRTOT,LSTOT,LD,LB,LPROF                             0002
     1 FORMAT( )                                                          0003
     2 FORMAT(A3)                                                         0004
     3 FORMAT(5A6)                                                        0005
       DATA IUSEA1,IUSEA2,IUSEB1/1,0,1/                                   0006
       DATA MCAN,MTIMES,NCODE,NTIME,NOUT,NOWN/1,1,4,1,0,0/                0007
       DATA NCR,NG1,NG2,NG3,NG4,NLP,NTP/8,12,13,14,15,16,5/               0008
C
C      READ IN DATA ***************************************************
C
       WRITE(NTP,5000)                                                    0009
 5000 FORMAT(/T2,'ANSWER ALL QUESTIONS YES OR NO.'/T2,'ALL INPUT IN FRE   0010
     1E FORMAT.',//T2,'DO YOU KNOW HOW TO USE THIS PROGRAM?')             0011
       READ(NCR,2) ANAME                                                  0012
       IF(ANAME.EQ.'YES') GO TO 501                                       0013
       CALL TEACH(NCODE)                                                  0014
       GO TO 902                                                          0015
  501 CONTINUE                                                            0016
       WRITE(NTP,5001)                                                    0017
 5001 FORMAT(/T2,'DO YOU WANT TO ADD A DATA ELEMENT?')                    0018
       READ(NCR,2) CNAME                                                  0019
       IF(CNAME.EQ.'NO') GO TO 502                                        0020
       MCAN=2                                                             0021
       WRITE(NTP,5002)                                                    0022
 5002 FORMAT(/T2,'ADD YOUR DATA ELEMENT.')                                0023
  502 CONTINUE                                                            0024
```

```
C
      CALL READIN(MCAN)                                                    0025
      IF(NJ.GT.999) GO TO 902                                              0026
C
C     'DO' LOOPS 51 AND 52 SET UP ARRAYS REQUIRED FOR WRITING DATA TO
C     FILE FOR PLOTTING PURPOSES.
      SUMTN=0.                                                             0027
      DO 51 J=1,NP2                                                        0028
      TRAYNO(J)=SUMTN                                                      0029
      SUMTN=SUMTN+1.                                                       0030
   51 CONTINUE                                                             0031
      SUMCN=0.                                                             0032
      DO 52 I=1,NC                                                         0033
      SUMCN=SUMCN+1.                                                       0034
      COMPNO(I)=SUMCN                                                      0035
   52 CONTINUE                                                             0036
 1115 CONTINUE                                                             0037
C
C     START OF RUN ***************************************************
C
C     READ RUN VARIABLES *********************************************
C
      ICOUNT=1                                                             0038
      WRITE(NTP,5011)                                                      0039
 5011 FORMAT(/T2,'TO PROCEED,TYPE GO OR TO STOP THE PROGRAM,TYPE END.')    0040
      READ(NCR,2) RUN                                                      0041
      IF(RUN.EQ.'END') GO TO 901                                           0042
      WRITE(NTP,5012)                                                      0043
 5012 FORMAT(/T2,'ENTER RUN NO. AS:  NO.,DAY,MONTH,  (INTEGER VALUES)')    0044
      READ(NCR,1) NUM1,NUM2,NUM3                                           0045
      IF(NUM2.GE.32.OR.NUM3.GE.13) GO TO 991                               0046
      IF(MTIMES.EQ.1) GO TO 515                                            0047
      WRITE(NTP,5013)                                                      0048
 5013 FORMAT(/T2,'ENTER REFLUX RATIO AND DISTILLATE RATE FOR NEW RUN.')    0049
      READ(NCR,1) RR,DTOT                                                  0050
```

```
      WRITE(NTP,5014)                                                    0051
 5014 FORMAT(T2,'ENTER FEED TEMPERATURE. (DEG C)')                       0052
      READ(NCR,1) TFEED                                                  0053
      NQVAL=1                                                            0054
      IF(NENTH.EQ.0) GO TO 514                                           0055
      WRITE(NTP,5015)                                                    0056
 5015 FORMAT(T2,'IS FEED PRESSURE GREATER THAN COLUMN PRESSURE?')        0057
      READ(NCR,2) FPRES                                                  0058
      IF(FPRES.EQ.'NO') GO TO 514                                        0059
      WRITE(NTP,5016)                                                    0060
 5016 FORMAT(T2,'ENTER FEED ENTHALPY. (KJ/KGMOLE)')                      0061
      READ(NCR,1) FENTH                                                  0062
      NQVAL=2                                                            0063
  514 CONTINUE                                                           0064
      WRITE(NTP,5017)                                                    0065
 5017 FORMAT(T2,'ENTER FEED TRAY POSN. (NO. FROM BOTTOM,REBOILER=0)')    0066
      READ(NCR,1) NF                                                     0067
      NF1=NF+1                                                           0068
      NF2=NF+2                                                           0069
  515 CONTINUE                                                           0070
      IF(RR.LE.0.) GO TO 991                                             0071
      WRITE(NTP,5111)                                                    0072
 5111 FORMAT(//T2,'START OF COMPUTATION.')                               0073
C     INITIAL TEMPERATURE PROFILE ***************************************
      DELTAT=(TBOT-TTOP)/NP1                                             0074
      DO 53 J=1,NP2                                                      0075
      TEMP(J)=TBOT-(J-1)*DELTAT                                          0076
   53 CONTINUE                                                           0077
C     TRAY PRESSURE PROFILE *********************************************
      IF(MTIMES.GT.1) GO TO 516                                          0078
      PRESS(NP1)=PCOND+DELTAC                                            0079
      PRESS(NP2)=PCOND                                                   0080
      NCOUNT=0                                                           0081
      DO 54 J=NP,1,-1                                                    0082
      NCOUNT=NCOUNT+1                                                    0083
      PRESS(J)=PRESS(NP1)+NCOUNT*DELTAP                                  0084
   54 CONTINUE                                                           0085
  516 CONTINUE                                                           0086
      MTIMES=MTIMES+1                                                    0087
```

```
C
C     FLASH CALCULATION TO DETERMINE FEED CONDITION ***************************
      CALL QFLASH(NTIME,TDEWPT)                                          0088
      IF(NJ.GT.999) GO TO 902                                            0089
C
C     INITIAL VAPOUR & LIQUID FLOW RATES IN COLUMN ****************************
      CALL FLOWEQ                                                        0090
      IF(NJ.GT.999) GO TO 902                                            0091
C
 1111 CONTINUE                                                           0092
C
C     START OF ITERATIONS FORCING THETA TO UNITY ******************************
C
C     CALCULATION OF EQUILIBRIUM CONSTANTS ************************************
      NEQ=3                                                              0093
      CALL EQUCON(NEQ)                                                   0094
C
C     ENTHALPY CONSIDERATIONS IGNORED FOR FIRST TWO ITERATIONS
      IF(ICOUNT.GT.2) GO TO 111                                          0095
      GO TO 112                                                          0096
C
C     ENTHALPY CALCULATIONS AND VAPOUR & LIQUID FLOWRATE CORRECTION ***********
  111 CONTINUE                                                           0097
      IF(NENTH.EQ.0) GO TO 112                                           0098
C     ENTHALPY CALCULATIONS ***************************************************
      NEN=3                                                              0099
      TDEWP=TDEWPT                                                       0100
      IF(TFEED.LT.TDEWPT) TDEWP=TFEED                                    0101
      IF(NOWN.EQ.1) CALL ENERGY(TBUBP,TDEWP,NEN,S1,S2,S3,S4)             0102
      IF(NOWN.EQ.2) CALL ENTHAL(ALIQ,AVAP,BM,DM,BTOT,DTOT,FTOT,HD,HFEED,  0103
     1HLIQ,HVAP,NC,NF1,NP1,NP2,NCOND,QC,QF,QR,TEMP,VRTOT,X,XF,Y,YF,TBUBP  0104
     2,TDEWP,TFEED,NEN,S1,S2,S3,S4,NTIME,MC,MP)                          0105
C     RECALCULATION OF FLOWRATES (NON-EQUIMOLAR OVERFLOW) *********************
      CALL FLOWNE                                                        0106
      IF(NJ.GT.999) GO TO 902                                            0107
  112 CONTINUE                                                           0108
```

```
C
C     ABSORPTION & STRIPPING FACTORS ***********************************************
      CALL ABSTRP                                                              0109
C
C     V(I)/B(I) & L(I)/B(I) RATIOS *************************************************
      CALL VBLB(LB,NC,NF1,S,VB,MC,MP)                                          0110
C
C     V(I)/D(I) & L(I)/D(I) RATIOS *************************************************
      CALL VDLD(A,LD,NC,NF2,NP1,NP2,VD,MC,MP)                                  0111
C
C     CALCULATED B(I)/D(I) RATIOS **************************************************
      CALL BDCALC(A,BD,D,FTOT,LB,NC,NF1,SUMDI,VD,Z,MC,MP)                      0112
      ERRD=DTOT-SUMDI                                                          0113
C
C     'THETA' CONVERGENCE TECHNIQUE ************************************************
      CALL THECAL                                                              0114
      WRITE(NTP,1000) THETA                                                    0115
 1000 FORMAT(T2,'THETA=',F12.4)                                                0116
      IF(ABS(THETA).GT.10000.) GO TO 115                                       0117
C
C     CHECK CONVERGENCE (THETA= 1.) ************************************************
      ERROR=1.-THETA                                                           0118
      IF(ABS(ERROR).LE.ACCLIM) GO TO 1112                                      0119
      IF(ABS(ERRD).LE.ACCLIM.AND.ABS(ERROR).LE.0.0001) GO TO 1112              0120
      GO TO 116                                                                0121
  115 CONTINUE                                                                 0122
      THETA=10000.                                                             0123
  116 CONTINUE                                                                 0124
C
C     CORRECTED B(I)/D(I) RATIOS AND B(I) & D(I) VALUES *****************************
      CALL BDCORR(B,BD,D,FTOT,NC,SLIQ,SUMD,SUMB,THETA,Z,MC,MP)                 0125
      DO 11 I=1,NC                                                             0126
      X(I,NP2)=D(I)/SUMD                                                       0127
      X(I,1)=B(I)/SUMB                                                         0128
   11 CONTINUE                                                                 0129
```

```
C
C       STRIPPING SECTION;  CORRECTED VAPOUR & LIQUID MOLAR FLOWRATES.
        CALL VLSTRP                                                     0130
C
C       RECTIFYING SECTION;  CORRECTED VAPOUR & LIQUID MOLAR FLOWRATES.
        CALL VLRECT                                                     0131
C
        NFDO=NP1                                                        0132
        IF(MEFF.EQ.0.OR.NCOND.GT.1) GO TO 117                           0133
        GO TO 118                                                       0134
  117   NP1=NP2                                                         0135
        GO TO 119                                                       0136
  118   CONTINUE                                                        0137
C       FOR INEFFICIENT SYSTEMS A TOTAL CONDENSER IS ASSUMED TO BE PERFECT.
C       FOR SUCH CASES THE CONDENSER TEMPERATURE IS CALCULATED HERE **************
        MMD=2                                                           0138
        CALL DEWCAL(NP2,MMD,TDEWP)                                      0139
        TD=TEMP(NP2)                                                    0140
        CALL BUBCAL(NP2,MMD,TBUBP)                                      0141
        TEMP(NP2)=(TD+TEMP(NP2))/2.                                     0142
C       NEW TEMPERATURE PROFILE THROUGH THE COLUMN ********************************
  119   CONTINUE                                                        0143
        DO 12 J=1,NP1                                                   0144
        MMD=1                                                           0145
        CALL DEWCAL(J,MMD,TDEWP)                                        0146
        TD=TEMP(J)                                                      0147
        CALL BUBCAL(J,MMD,TBUBP)                                        0148
        TEMP(J)=(TD+TEMP(J))/2.                                         0149
   12   CONTINUE                                                        0150
        NP1=NFDO                                                        0151
        ICOUNT=ICOUNT+1                                                 0152
C       MAXIMUM OF 30 ITERATIONS ALLOWED FOR CONVERGENCE.
        IF(ICOUNT.GE.30) GO TO 1112                                     0153
C       RETURN FOR NEXT ITERATION *************************************************
        GO TO 1111                                                      0154
 1112   CONTINUE                                                        0155
        WRITE(NTP,5112)                                                 0156
 5112   FORMAT(/T2,'COMPUTATION COMPLETE.',/)                           0157
```

```
C     WRITE RESULTS TO TERMINAL ***************************************************
      CALL RUNRES                                                              0158
C
C     RETURN FOR NEXT RUN WHEN NO DETAILED RESULTS ARE REQUIRED ***************
      IF(NOUT.GE.5) GO TO 1115                                                 0159
      WRITE(NTP,5018)                                                          0160
 5018 FORMAT(//T2,'DO YOU WANT A FULL PRINT-OUT?')                             0161
      READ(NCR,2) PROUT                                                        0162
      IF(PROUT.EQ.'NO') GO TO 1115                                             0163
      NOUT=NOUT+1                                                              0164
      IF(NOUT.EQ.5) WRITE(NTP,8091)                                            0165
 8091 FORMAT(T2,'THIS IS THE LAST FULL PRINT-OUT FOR THIS EXECUTION.')         0166
C
C     PRINTOUT OF DETAILED RESULTS ************************************************
      CALL OUTPUT(ERROR,ERRD)                                                  0167
C
C     WRITE TO FILE FOR GRAPHICAL OUTPUT ******************************************
 8080 FORMAT(A4)                                                               0168
 8081 FORMAT(3I4)                                                              0169
 8082 FORMAT(2F10.2)                                                           0170
 8083 FORMAT(F10.2,F7.4)                                                       0171
      IUSEA2=IUSEA2+2                                                          0172
      VPROF(NP2)=DTOT                                                          0173
      PRD='DATA'                                                               0174
      WRITE(NG1,8080) PRD                                                      0175
      WRITE(NG1,8081) NP2,IUSEA1,IUSEA2                                        0176
      WRITE(NG1,8082) (TEMP(J),TRAYNO(J),J=NP2,1,-1)                           0177
      WRITE(NG2,8080) PRD                                                      0178
      WRITE(NG2,8081) NP2,IUSEB1,IUSEA2                                        0179
      WRITE(NG2,8082) (LPROF(J),TRAYNO(J),J=NP2,1,-1)                          0180
      IUSEB2=IUSEA2-1                                                          0181
      WRITE(NG2,8080) PRD                                                      0182
      WRITE(NG2,8081) NP2,IUSEB1,IUSEB2                                        0183
      WRITE(NG2,8082) (VPROF(J),TRAYNO(J),J=NP2,1,-1)                          0184
      WRITE(NG3,8080) PRD                                                      0185
      WRITE(NG3,8081) NC,IUSEA1,IUSEA2                                         0186
      WRITE(NG3,8083) (COMPNO(I),X(I,NP2),I=1,NC)                              0187
```

```
      IUSE1=1                                                          0188
      IUSE3=1                                                          0189
      DO 86 I=1,NC                                                     0190
      WRITE(NG4,8080) PRD                                              0191
      WRITE(NG4,8081) NP2,IUSE1,IUSE3                                  0192
      WRITE(NG4,8083) (TRAYNO(J),X(I,J),J=NP2,1,-1)                    0193
      IUSE3=IUSE3+1                                                    0194
   86 CONTINUE                                                         0195
C     RETURN FOR NEXT RUN AFTER WRITING FULL RESULTS TO FILES **********
      GO TO 1115                                                       0196
C
C     ERROR MESSAGES AND TERMINATION ROUTINES **************************
  991 CONTINUE                                                         0197
      WRITE(NTP,9911)                                                  0198
 9911 FORMAT(/T2,'DATA NOT ACCEPTABLE.')                               0199
      WRITE(NTP,9912)                                                  0200
 9912 FORMAT(/T2,'ILLEGAL SPECIFICATION ENCOUNTERED',/T2,              0201
     1'REFER TO THE USER MANUAL FOR THE CORRECT PROCEDURE.',/T2,       0202
     2'AND START THE PROGRAM AGAIN.')                                 0203
      CALL ERTRAN(6,'@ADD DISTFILE,FINISH . ')                         0204
      GO TO 999                                                        0205
  901 CONTINUE                                                         0206
      WRITE(NTP,9901)                                                  0207
 9901 FORMAT(/T2,'SUCCESSFUL EXECUTION COMPLETED.')                    0208
      IF(NOUT.LT.1) GO TO 902                                          0209
      WRITE(NTP,9902)                                                  0210
 9902 FORMAT(/T2,'TO SEND FULL PRINT-OUT TO LINE PRINTER,ENTER THE',   0211
     1/T2,'FOLLOWING COMMANDS:',/T2,'@ADD DISTFILE,PRT',/T2,           0212
     2'@ED,U PRINTOUT.',/T2,'PRINT!',/T2,'OMIT',/T2,'@ADD DISTFILE,PR2')  0213
      WRITE(NTP,9903)                                                  0214
 9903 FORMAT(/T2,'COLLECT FULL PRINT-OUT FROM COMPUTER CENTRE.')       0215
  902 CONTINUE                                                         0216
      IF(NOUT.LT.1) CALL ERTRAN(6,'@ADD DISTFILE.FINISH . ')           0217
  999 CONTINUE                                                         0218
      WRITE(NTP,9999)                                                  0219
 9999 FORMAT(/T2,'END OF PROGRAM.',//)                                 0220
      STOP                                                             0221
      END                                                              0222
```

```
      SUBROUTINE READIN(MCAN)                                                  0001
C     ROUTINE FOR READING DATA FROM EITHER FILE OR TERMINAL,FOR
C     MULTICOMPONENT DISTILLATION PROBLEMS.
C     ****************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ****************************************************
      DIMENSION F1(5),F2(5),F3(5),F4(5),U1(4),U2(4),U3(4),U4(4)                0002
      REAL K,LMFR,LRTOT,LSTOT,LD,LB,LPROF                                      0003
      DATA F1/'@ASG,UP             .,F17  .     '/                             0004
      DATA F2/'@ASG,UP             .,F17  .     '/                             0005
      DATA F3/'@ASG,UP             .,F17  .     '/                             0006
      DATA F4/'@ASG,UP             .,F17  .     '/                             0007
      DATA U1/'@USE     12,        .  .  '/                                    0008
      DATA U2/'@USE     13,        .  .  '/                                    0009
      DATA U3/'@USE     14,        .  .  '/                                    0010
      DATA U4/'@USE     15,        .  .  '/                                    0011
    1 FORMAT( )                                                                0012
    2 FORMAT(A3)                                                               0013
    3 FORMAT(5A6)                                                              0014
    4 FORMAT(4(A6,1X))                                                         0015
      DATA ATM,FACTP1,FACTP2/14.696,51.714752,7.50061683/                      0016
      IF(MCAN.EQ.1) WRITE(NTP,1001)                                            0017
 1001 FORMAT(T2,'ENTER ALL DATA IN FREE FORMAT,',/T2,'ENTER NO. OF COMPO       0018
     1NENTS IN FEED.')                                                         0019
      READ(NCR,1) NC                                                           0020
      IF(MCAN.EQ.1) WRITE(NTP,1002)                                            0021
 1002 FORMAT(T2,'ENTER NO. OF TRAYS IN COLUMN,',/T2,'EXCLUDE CONDENSER A       0022
     1ND REBOILER,')                                                           0023
      READ(NCR,1) NP                                                           0024
```

```
      IF(MCAN.EQ.1) WRITE(NTP,1003)                                          0025
 1003 FORMAT(T2,'ENTER FEED TRAY POSN. (NO. FROM BOTTOM,REBOILER=0),')       0026
      READ(NCR,1) NF                                                         0027
      NP1=NP+1                                                               0028
      NP2=NP+2                                                               0029
      NF1=NF+1                                                               0030
      NF2=NF+2                                                               0031
      IF(NC.GT.MC.OR.NP2.GE.MP) GO TO 902                                    0032
      IF(MCAN.EQ.1) WRITE(NTP,1004)                                          0033
 1004 FORMAT(T4,'ACCEPTED PRESSURE UNITS: 1=MMHG,2=PSIG,3=PSIA,4=KPA,5=A     0034
     1TM.',/T2,'ENTER CONDENSER PRESSURE AND UNIT CODE. (EG, 760.0,1)')      0035
      READ(NCR,1) PCONDA,NUNITP                                              0036
      IF(NUNITP.EQ.1) PCOND=PCONDA                                           0037
      IF(NUNITP.EQ.2) PCOND=(PCONDA+ATM)*FACTP1                              0038
      IF(NUNITP.EQ.3) PCOND=PCONDA*FACTP1                                    0039
      IF(NUNITP.EQ.4) PCOND=PCONDA*FACTP2                                    0040
      IF(NUNITP.EQ.5) PCOND=PCONDA*ATM*FACTP1                                0041
      IF(MCAN.EQ.1) WRITE(NTP,1005)                                          0042
 1005 FORMAT(T4,'ACCEPTED PRESSURE DROP UNITS: 1=MMHG,2=PSI,3=KPA',          0043
     1/T2,'ENTER PRESSURE DROP PER TRAY AND UNIT CODE.')                     0044
      READ(NCR,1) DELTPD,NUNIPD                                              0045
      IF(NUNIPD.EQ.1) DELTAP=DELTPD                                          0046
      IF(NUNIPD.EQ.2) DELTAP=DELTPD*FACTP1                                   0047
      IF(NUNIPD.EQ.3) DELTAP=DELTPD*FACTP2                                   0048
      IF(MCAN.EQ.1) WRITE(NTP,1105)                                          0049
 1105 FORMAT(T2,'ENTER PRESSURE DROP ACROSS CONDENSER AND UNIT CODE.')       0050
      READ(NCR,1) DELTPC,NUNIPC                                              0051
      IF(NUNIPC.EQ.1) DELTAC=DELTPC                                          0052
      IF(NUNIPC.EQ.2) DELTAC=DELTPC*FACTP1                                   0053
      IF(NUNIPC.EQ.3) DELTAC=DELTPC*FACTP2                                   0054
```

```
      IF(MCAN.EQ.1) WRITE(NTP,1006)                                     0055
 1006 FORMAT(T4,'ACCEPTED TEMPERATURE UNITS: 1=DEG C,2=DEG F',/T2,       0056
     1'ENTER ESTIMATE OF REBOILER AND CONDENSER TEMPERATURES WITH UNIT C  0057
     2ODES.',/T2,'(EG. 100.0,1.50.0,2)')                                 0058
      READ(NCR,1) TBOTG,NUTB,TTOPG,NUTT                                  0059
      IF(NUTB.EQ.1) TBOT=TBOTG                                           0060
      IF(NUTB.EQ.2) TBOT=(TBOTG-32.)/1.8                                 0061
      IF(NUTT.EQ.1) TTOP=TTOPG                                           0062
      IF(NUTT.EQ.2) TTOP=(TTOPG-32.)/1.8                                 0063
      IF(MCAN.EQ.1) WRITE(NTP,1007)                                      0064
 1007 FORMAT(T2,'ENTER NAMES OF COMPONENTS. (ONE PER LINE,MAX 30 CHARACT  0065
     1ERS EACH.)')                                                       0066
      READ(NCR,3) ((TITLES(I,J),J=1,5),I=1,NC)                           0067
      IF(MCAN.EQ.1) WRITE(NTP,1008)                                      0068
 1008 FORMAT(T2,'ENTER FEED RATE.  (KG MOLES/HR.)')                      0069
      READ(NCR,1) FTOT                                                   0070
      IF(MCAN.EQ.1) WRITE(NTP,1009)                                      0071
 1009 FORMAT(T2,'ENTER FEED STREAM COMPOSITIONS, (MOL FRACTIONS)')        0072
      READ(NCR,1) (Z(I),I=1,NC)                                          0073
      IF(MCAN.EQ.1) WRITE(NTP,2110)                                      0074
 2110 FORMAT(T2,'ENTER FEED TEMPERATURE. (DEG C)')                       0075
      READ(NCR,1) TFEED                                                  0076
      NQVAL=1                                                            0077
      NENTH=10                                                           0078
      IF(MCAN.EQ.1) WRITE(NTP,1011)                                      0079
 1011 FORMAT(T2,'ARE ENTHALPY EFFECTS TO BE CONSIDERED?')                0080
      READ(NCR,2) ENTHYP                                                 0081
      IF(ENTHYP.NE.'YES') NENTH=0                                        0082
      IF(NENTH.EQ.0) GO TO 113                                           0083
      IF(MCAN.EQ.1) WRITE(NTP,2111)                                      0084
 2111 FORMAT(T2,'IS FEED PRESSURE GREATER THAN COLUMN PRESSURE?')        0085
      READ(NCR,2) FPRES                                                  0086
```

```
      IF(FPRES.EQ.'NO') GO TO 113                                        0087
      IF(MCAN.EQ.1) WRITE(NTP,2112)                                      0088
 2112 FORMAT(T2,'ENTER FEED ENTHALPY, (KJ/KGMOLE)')                      0089
      READ(NCR,1) FENTH                                                  0090
      NQVAL=2                                                            0091
  113 CONTINUE                                                           0092
      IF(MCAN.EQ.1) WRITE(NTP,1010)                                      0093
 1010 FORMAT(T4,'TOTAL OR PARTIAL CONDENSER?',/T2,'ENTER TOT OR PAR.')   0094
  110 CONTINUE                                                           0095
      READ(NCR,2) CONTYP                                                 0096
      IF(CONTYP.NE.'TOT'.AND.CONTYP.NE.'PAR') GO TO 901                  0097
      IF(CONTYP.EQ.'TOT') GO TO 111                                      0098
      NCOND=2                                                            0099
      GO TO 112                                                          0100
  111 CONTINUE                                                           0101
      NCOND=1                                                            0102
  112 CONTINUE                                                           0103
      IF(ENTHYP.NE.'YES') GO TO 120                                      0104
      IF(MCAN.EQ.1) WRITE(NTP,1012)                                      0105
 1012 FORMAT(T2,'DO YOU WANT TO USE YOUR OWN ENTHALPY ROUTINE?')         0106
      READ(NCR,2) OWNENT                                                 0107
      IF(OWNENT.EQ.'YES') GO TO 114                                      0108
      IF(MCAN.EQ.1) WRITE(NTP,1013)                                      0109
 1013 FORMAT(T4,'REFER TO MANUAL,SECTION 4,FOR ENTHALPY ROUTINE CODES,'  0110
     1/T2,'ENTER CODE NO. OF LIBRARY ENTHALPY ROUTINE REQUIRED.')        0111
      READ(NCR,1) NENTH                                                  0112
      NOWN=1                                                             0113
      IF(MCAN.EQ.1) WRITE(NTP,1113)                                      0114
 1113 FORMAT(T2,'ENTER ENTHALPY DATA AS DESCRIBED IN THE MANUAL,SECTION  0115
     14,'./T2,'OR ADD A DATA ELEMENT.')                                  0116
      READ(NCR,1) (HL1(I),HL2(I),HL3(I),I=1,NC)                          0117
      READ(NCR,1) (HV1(I),HV2(I),HV3(I),I=1,NC)                          0118
      GO TO 120                                                          0119
  114 CONTINUE                                                           0120
```

```
      IF(MCAN.EQ.1) WRITE(NTP,1014)                                      0121
 1014 FORMAT(T2,'IT IS ASSUMED THAT THE ROUTINE CONFORMS TO SPECIFICATIO  0122
     1NS',/T2,'SET OUT IN SECTION 4 IN THE MANUAL.')                     0123
      NENTH=5                                                            0124
      NOWN=2                                                             0125
  120 CONTINUE                                                           0126
      IF(MCAN.EQ.1) WRITE(NTP,1018)                                      0127
 1018 FORMAT(T2,'ENTER EQUIL. DATA AS DESCRIBED IN THE MANUAL,SECTION 4,  0128
     1',/T2,'OR ADD A DATA ELEMENT.')                                    0129
      READ(NCR,1) NEQUIL                                                 0130
      IF(NEQUIL.EQ.1) GO TO 131                                          0131
      IF(NEQUIL.EQ.3) GO TO 132                                          0132
      GO TO 133                                                          0133
  131 CONTINUE                                                           0134
      READ(NCR,1) (AN(I),BN(I),CN(I),I=1,NC)                             0135
      GO TO 134                                                          0136
  132 CONTINUE                                                           0137
      READ(NCR,1) AKB1,AKB2,AKB3,AKB4                                    0138
  133 CONTINUE                                                           0139
      READ(NCR,1) (AK1(I),AK2(I),AK3(I),AK4(I),I=1,NC)                   0140
      IF(NEQUIL.NE.3) GO TO 134                                          0141
      DO 10 I=1,NC                                                       0142
      IF(ABS(AK2(I)).LT.0.00001) NJ=I                                    0143
   10 CONTINUE                                                           0144
  134 CONTINUE                                                           0145
      IF(MCAN.EQ.1) WRITE(NTP,1019)                                      0146
 1019 FORMAT(T2,'DO YOU WANT TO SPECIFY THE THETA COVERGENCE LIMIT?')     0147
      READ(NCR,2) CONLIM                                                 0148
      IF(CONLIM.EQ.'YES') GO TO 135                                      0149
      ACCLIM=0.00005                                                     0150
      GO TO 136                                                          0151
  135 CONTINUE                                                           0152
      IF(MCAN.EQ.1) WRITE(NTP,1020)                                      0153
 1020 FORMAT(T2,'ENTER CONVERGENCE LIMIT.')                              0154
      READ(NCR,1) ACCLIM                                                 0155
  136 CONTINUE                                                           0156
```

```
      IF(MCAN.EQ.1) WRITE(NTP,1021)                                    0157
 1021 FORMAT(T2,'ARE PERFECT TRAYS ASSUMED?')                          0158
      EFFVAP=1.0                                                       0159
      MEFF=0                                                           0160
      READ(NCR,2) EFFCRI                                               0161
      IF(EFFCRI.EQ.'YES') GO TO 137                                    0162
      IF(MCAN.EQ.1) WRITE(NTP,1022)                                    0163
 1022 FORMAT(T2,'ADD EFFICIENCY DATA AS DESCRIBED IN THE MANUAL,SECTION 0164
     14.')                                                             0165
      READ(NCR,1) MEFF                                                 0166
      IF(MEFF.EQ.2) GO TO 138                                          0167
      READ(NCR,1) ((EFFV(I,J),I=1,NC),J=NP2,1,-1)                      0168
      GO TO 139                                                        0169
  138 CONTINUE                                                         0170
      READ(NCR,1) (EFFV(I,1),I=1,NC)                                   0171
      DO 11 I=1,NC                                                     0172
      DO 12 J=2,NP2                                                    0173
      EFFV(I,J)=EFFV(I,1)                                              0174
   12 CONTINUE                                                         0175
   11 CONTINUE                                                         0176
      GO TO 139                                                        0177
  137 CONTINUE                                                         0178
      DO 13 J=1,NP2                                                    0179
      DO 14 I=1,NC                                                     0180
      EFFV(I,J)=EFFVAP                                                 0181
   14 CONTINUE                                                         0182
   13 CONTINUE                                                         0183
  139 CONTINUE                                                         0184
      IF(MCAN.EQ.1) WRITE(NTP,1024)                                    0185
 1024 FORMAT(T2,'ENTER REFLUX RATIO AND DISTILLATE WITHDRAWAL RATE.')   0186
      READ(NCR,1) RR,DTOT                                              0187
```

```
      WRITE(NTP,1025)                                                    0188
 1025 FORMAT(T2,'DO YOU WANT GRAPHICAL OUTPUT?  (SEE MANUAL SECTION 4)') 0189
      READ(NCR,2) USEF1                                                  0190
      IF(USEF1.NE.'YES') GO TO 140                                       0191
      WRITE(NTP,1026)                                                    0192
 1026 FORMAT(T2,'DO YOU KNOW HOW TO USE THE PLOT ROUTINE?')              0193
      READ(NCR,2) USEF2                                                  0194
      IF(USEF2.NE.'YES') GO TO 140                                       0195
      WRITE(NTP,1027)                                                    0196
 1027 FORMAT(T2,'ENTER DATA-FILE NAMES.  (SEE MANUAL SECTION 4)')        0197
      READ(NCR,4) F1(3),F2(3),F3(3),F4(3)                                0198
      U1(3)=F1(3)                                                        0199
      U2(3)=F2(3)                                                        0200
      U3(3)=F3(3)                                                        0201
      U4(3)=F4(3)                                                        0202
      CALL ERTRAN(6,F1)                                                  0203
      CALL ERTRAN(6,F2)                                                  0204
      CALL ERTRAN(6,F3)                                                  0205
      CALL ERTRAN(6,F4)                                                  0206
      CALL ERTRAN(6,U1)                                                  0207
      CALL ERTRAN(6,U2)                                                  0208
      CALL ERTRAN(6,U3)                                                  0209
      CALL ERTRAN(6,U4)                                                  0210
  140 CONTINUE                                                           0211
      GO TO 999                                                          0212
C     ERROR MESSAGES ***************************************************
  901 CONTINUE                                                           0213
      IF(MCAN.EQ.1) WRITE(NTP,9001)                                      0214
 9001 FORMAT(T2,'ONLY TOT OR PAR IS ACCEPTABLE.',/T2,'RETYPE NOW.')      0215
      GO TO 110                                                          0216
  902 CONTINUE                                                           0217
      WRITE(NTP,9002)                                                    0218
 9002 FORMAT(/T2,'DIMENSION SPECIFICATIONS EXCEEDED.')                   0219
      NJ=1000                                                            0220
  999 CONTINUE                                                           0221
      RETURN                                                             0222
      END                                                                0223
```

```
      SUBROUTINE EQUCON(NEQ)                                              0001
C     CALCULATION OF EQUILIBRIUM CONSTANTS.
C     CHOICE OF FOUR PROCEDURES MADE BY COUNTER NEQUIL
C       (1) USE OF ANTOINE'S EQUATION,HENRY'S LAW AND RAOULT'S LAW.
C           IE.  LOG10(PVAP)=AN-BN/(CN+T),Y=K*X,Y=PVAP*X/PT
C                PV=VAPOUR PRESS,PT=TOTAL PRESS,T IN DEG C
C                A,B,C ARE ANTOINE CONSTS FOR COMPONENT
C       (2) EQUILIBRIUM CONSTANT EXPRESSED AS A FUNCTION OF TEMPERATURE
C           IE.  K=EXP(AK1-AK2/T+AK3*T)      T IN DEG R
C       (3) RELATIVE VOLATILITY EXPRESSED AS A FUNCTION OF TEMPERATURE
C           WITH A KEY COMPONENT EQUIL CONST AS A FUNCTION OF T,AND THE
C           EQUATION:      REL VOL=K(I)/AKB  (AKB=KEY COMP EQUIL CONST)
C           FUNCTIONS ARE: REL VOL=AK1+AK2*T+AK3*T**2        T IN DEG F
C                  AKB=AKB1+AKB2*T+AKB3*T**2+AKB4*T**3       T IN DEG F
C       (4) EQUILIBRIUM CONSTANT EXPRESSED AS A FUNCTION OF TEMPERATURE
C           IE.  K=T*(AK1+AK2*T+AK3*T**2+AK4*T**3)**3        T IN DEG R
C     ****************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ****************************************************
      REAL K,KF,LMFR,LRTOT,LSTOT,LD,LB,LPROF                              0002
      GO TO (100,200,300,400),NEQUIL                                      0003
  100 CONTINUE                                                            0004
      IF(NEQ.EQ.1) GO TO 110                                              0005
C     CALCULATION USING OPTION (1) ABOVE.
      DO 11 J=1,NP2                                                       0006
      DO 12 I=1,NC                                                        0007
      PVAP=10.**(AN(I)-BN(I)/(CN(I)+TEMP(J)))                             0008
      K(I,J)=PVAP*EFFV(I,J)/PRESS(J)                                      0009
   12 CONTINUE                                                            0010
   11 CONTINUE                                                            0011
  110 CONTINUE                                                            0012
      DO 13 I=1,NC                                                        0013
      PVAP=10.**(AN(I)-BN(I)/(CN(I)+TFEED))                               0014
      KF(I)=PVAP/PRESS(NF1)                                               0015
   13 CONTINUE                                                            0016
      GO TO 999                                                           0017
```

```
  200 CONTINUE                                                          0018
      IF(NEQ.EQ.1) GO TO 210                                            0019
C     CALCULATION USING OPTION (2) ABOVE.
      DO 21 J=1,NP2                                                     0020
      T1=TEMP(J)*1.8+32.+459.4                                          0021
      T=T1/100.                                                         0022
      DO 22 I=1,NC                                                      0023
      K(I,J)=EXP(AK1(I)-AK2(I)/T+AK3(I)*T)*EFFV(I,J)                    0024
   22 CONTINUE                                                          0025
   21 CONTINUE                                                          0026
  210 CONTINUE                                                          0027
      T1=TFEED*1.8+32.+459.4                                            0028
      T=T1/100.                                                         0029
      DO 23 I=1,NC                                                      0030
      KF(I)=EXP(AK1(I)-AK2(I)/T+AK3(I)*T)                               0031
   23 CONTINUE                                                          0032
      GO TO 999                                                         0033
  300 CONTINUE                                                          0034
      IF(NEQ.EQ.1) GO TO 310                                            0035
C     CALCULATION USING OPTION (3) ABOVE.
C     CALCULATION USING OPTION (3) ABOVE.
      DO 31 J=1,NP2                                                     0036
      T1=TEMP(J)*1.8+32.                                                0037
      T=T1/100.                                                         0038
      AKB=AKB1+AKB2*T+AKB3*T**2.+AKB4*T**3.                             0039
      DO 32 I=1,NC                                                      0040
      K(I,J)=(AK1(I)+AK2(I)*T+AK3(I)*T**2.)*AKB*EFFV(I,J)               0041
   32 CONTINUE                                                          0042
   31 CONTINUE                                                          0043
  310 CONTINUE                                                          0044
      T1=TFEED*1.8+32.                                                  0045
      T=T1/100.                                                         0046
      AKB=AKB1+AKB2*T+AKB3*T**2.+AKB4*T**3.                             0047
      DO 33 I=1,NC                                                      0048
      KF(I)=(AK1(I)+AK2(I)*T+AK3(I)*T**2.)*AKB                          0049
   33 CONTINUE                                                          0050
      GO TO 999                                                         0051
```

```
  400 CONTINUE                                                          0052
      IF(NEQ.EQ.1) GO TO 410                                            0053
C     CALCULATION USING OPTION (4) ABOVE.
      DO 41 J=1,NP2                                                     0054
      T1=TEMP(J)*1.8+32.+459.4                                          0055
      T=T1/1000.                                                        0056
      DO 42 I=1,NC                                                      0057
      AK=T1*(AK1(I)+AK2(I)*T+AK3(I)*T**2.+AK4(I)*T**3.)**3.             0058
      K(I,J)=AK*EFFV(I,J)                                               0059
   42 CONTINUE                                                          0060
   41 CONTINUE                                                          0061
  410 CONTINUE                                                          0062
      T1=TFEED*1.8+32.+459.4                                            0063
      T=T1/1000.                                                        0064
      DO 43 I=1,NC                                                      0065
      KF(I)=T1*(AK1(I)+AK2(I)*T+AK3(I)*T**2.+AK4(I)*T**3.)**3.          0066
   43 CONTINUE                                                          0067
  999 CONTINUE                                                          0068
      RETURN                                                            0069
      END                                                               0070
```

SUBROUTINE EQUCON

```
      SUBROUTINE ENERGY(TBUB,TDEW,NEN,S1,S2,S3,S4)                      0001
C     CALCULATION OF VAPOUR AND LIQUID ENTHALPY DATA,FEED ENTHALPY,
C     CONDENSER DUTY AND REBOILER DUTY.
C     CHOICE OF THREE AVAILABLE FUNCTIONS MADE BY NENTH.
C     FUNCTIONS ARE:
C       (A)  H=(A+B*T+C*T**2)**2      TI IN DEG R    (NENTH=1)
C       (B)  H=A+B*T+C*T**2           TI IN DEG F    (NENTH=2)
C       (C)  H=A+B*T+C*T**2           TI IN DEG R    (NENTH=3)
C     NOTE: T=TI/100. AND TI IS TEMPERATURE ON TRAY N
C           H = VAPOUR OR LIQUID MOLAR ENTHALPY FOR COMPONENT.
C     DIFFERENT COEFFICIENTS ARE NEEDED FOR EACH PHASE.
C     ************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ************************************************
      REAL K,KF,LMFR,LRTOT,LSTOT,LD,LB,LPROF                            0002
      DATA CONFAC,ENCONV/0.00027778,2.326/                              0003
      TRANK=0.0                                                         0004
      POWER=1.                                                          0005
      IF(NENTH.EQ.1) POWER=2.                                           0006
      IF(NENTH.EQ.1.OR.NENTH.EQ.3) TRANK=459.4                          0007
      IF(NEN.EQ.1) GO TO 200                                            0008
      IF(NEN.EQ.2) GO TO 103                                            0009
C     CALCULATION OF ENTHALPY OF VAPOUR AND LIQUID STREAMS.
      DO 10 J=1,NP2                                                     0010
      TI=TEMP(J)*1.8+32.+TRANK                                          0011
      T=TI/100.                                                         0012
      SUMHV=0.                                                          0013
      SUMHL=0.                                                          0014
      DO 11 I=1,NC                                                      0015
      HV=(HV1(I)+HV2(I)*T+HV3(I)*T**2.)**POWER*Y(I,J)                   0016
      HL=(HL1(I)+HL2(I)*T+HL3(I)*T**2.)**POWER*X(I,J)                   0017
      SUMHV=SUMHV+HV                                                    0018
      SUMHL=SUMHL+HL                                                    0019
   11 CONTINUE                                                          0020
      HVAP(J)=SUMHV*ENCONV                                              0021
      HLIQ(J)=SUMHL*ENCONV                                              0022
   10 CONTINUE                                                          0023
```

```
C       CALCULATION OF CONDENSER DUTY.
        HD=HLIQ(NP2)                                                    0024
        IF(NCOND.GT.1) GO TO 102                                        0025
C       TOTAL CONDENSER OPERATION.
        QC=VRTOT(NP1)*(HVAP(NP1)-HLIQ(NP2))                             0026
        GO TO 103                                                       0027
  102 CONTINUE                                                          0028
C       PARTIAL CONDENSER OPERATION.
        QC=VRTOT(NP1)*HVAP(NP1)-LRTOT(NP2)*HLIQ(NP2)-DTOT*HVAP(NP2)     0029
        HD=HVAP(NP2)                                                    0030
  103 CONTINUE                                                          0031
C       CALCULATION OF FEED STREAM ENTHALPY.
        SUMFL=0.                                                        0032
        SUMFV=0.                                                        0033
        TI=TFEED*1.8+32.+TRANK                                          0034
        T=TI/100.                                                       0035
        DO 12 I=1,NC                                                    0036
        HV=(HV1(I)+HV2(I)*T+HV3(I)*T**2.)**POWER*YF(I)                  0037
        HL=(HL1(I)+HL2(I)*T+HL3(I)*T**2.)**POWER*XF(I)                  0038
        SUMFV=SUMFV+HV                                                  0039
        SUMFL=SUMFL+HL                                                  0040
   12 CONTINUE                                                          0041
        HFEED=(AVAP/FTOT*SUMFV+ALIQ/FTOT*SUMFL)*ENCONV                  0042
        IF(NEN.EQ.2) GO TO 999                                          0043
        QFTOT=FTOT*HFEED                                                0044
C       CALCULATION OF REBOILER DUTY.
        QR=DTOT*HD+BTOT*HLIQ(1)+QC-QFTOT                                0045
        DM=HD+QC/DTOT                                                   0046
        BM=HLIQ(1)-QR/BTOT                                              0047
C       CONVERSION TO S.I. UNITS
        QC=QC*CONFAC                                                    0048
        QR=QR*CONFAC                                                    0049
        QF=HFEED                                                        0050
        GO TO 999                                                       0051
```

```
  200 CONTINUE                                                          0052
C     CALCULATION OF ENTHALPY FOR Q VALUE DETERMINATION.
      T1=TFEED*1.8+32.+TRANK                                            0053
      T=T1/100.                                                         0054
      TB=(TBUB*1.8+32.+TRANK)/100.                                      0055
      TD=(TDEW*1.8+32.+TRANK)/100.                                      0056
      S1=0.                                                             0057
      S2=0.                                                             0058
      S3=0.                                                             0059
      S4=0.                                                             0060
      DO 21 I=1,NC                                                      0061
      HF1=((HV1(I)+HV2(I)*T+HV3(I)*T**2.)**POWER*Z(I))*ENCONV           0062
      HF2=((HL1(I)+HL2(I)*T+HL3(I)*T**2.)**POWER*Z(I))*ENCONV           0063
      HV=((HV1(I)+HV2(I)*TD+HV3(I)*TD**2.)**POWER*Z(I))*ENCONV          0064
      HL=((HL1(I)+HL2(I)*TB+HL3(I)*TB**2.)**POWER*Z(I))*ENCONV          0065
      S1=S1+HF1                                                         0066
      S2=S2+HF2                                                         0067
      S3=S3+HV                                                          0068
      S4=S4+HL                                                          0069
   21 CONTINUE                                                          0070
  999 CONTINUE                                                          0071
      RETURN                                                            0072
      END                                                               0073
```

```
      SUBROUTINE QFLASH(NTIME,TDEWPT)                                    0001
C     CALCULATION OF COMPOSITION AND FLOWRATE OF THE VAPOUR AND LIQUID
C     FRACTIONS OF THE FEED STREAM.
C     THE SUBROUTINE CALCULATES THE FEED STREAM Q VALUE FOR EITHER FEED
C     TEMPERATURE OR FEED ENTHALPY SPECIFICATIONS.
C     IF THE SPECIFIED TEMPERATURE DOES NOT LIE WITHIN THE BUBBLE PT-
C     DEW PT RANGE,AND ENTHALPY DATA ARE NOT SUPPLIED THEN A Q VALUE
C     MUST BE GIVEN.
C     THE REQUIRED PROCEDURE IS INDICATED BY THE COUNTER NQVAL.
C     ********************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ********************************************************
      REAL K,KF,LMFR,LRTOT,LSTOT,LD,LB,LPROF                             0002
    1 FORMAT( )                                                          0003
      NTYP=1                                                             0004
      NVAP=NCOND                                                         0005
      SUMI=1.                                                            0006
      IT=NP2+1                                                           0007
      DO 10 I=1,NC                                                       0008
      XF(I)=Z(I)                                                         0009
      YF(I)=Z(I)                                                         0010
      X(I,IT)=Z(I)                                                       0011
      Y(I,IT)=Z(I)                                                       0012
   10 CONTINUE                                                           0013
      PRESS(IT)=PRESS(NF1)                                               0014
      TEMP(IT)=TFEED                                                     0015
      MD=2                                                               0016
      CALL BUBCAL(IT,MD,TBUBPT)                                          0017
      CALL DEWCAL(IT,MD,TDEWPT)                                          0018
      GO TO (100,300),NQVAL                                              0019
```

```
  100 CONTINUE                                                          0020
C     ISOBARIC FLASH CALCULATION.
      IF(TFEED.LT.TBUBPT.OR.TFEED.GT.TDEWPT) GO TO 120                  0021
      Q=0.01                                                            0022
      DELQ=0.001                                                        0023
      MN=1                                                              0024
      NEQ=1                                                             0025
      CALL EQUCON(NEQ)                                                  0026
  110 CONTINUE                                                          0027
      SUM=0.                                                            0028
      DO 11 I=1,NC                                                      0029
      XF(I)=Z(I)/(KF(I)-Q*KF(I)+Q)                                      0030
      SUM=SUM+XF(I)                                                     0031
   11 CONTINUE                                                          0032
      IF(ABS(SUM-SUM1).LT.0.0001) GO TO 111                             0033
      NCALL=2                                                           0034
      CALL SEARCH(SUM,SUM1,Q,DELQ,MN,NCALL)                             0035
      GO TO 110                                                         0036
  111 CONTINUE                                                          0037
      SUMY=0.                                                           0038
      DO 12 I=1,NC                                                      0039
      YF(I)=XF(I)*KF(I)                                                 0040
      SUMY=SUMY+YF(I)                                                   0041
   12 CONTINUE                                                          0042
      DO 13 I=1,NC                                                      0043
      YF(I)=YF(I)/SUMY                                                  0044
   13 CONTINUE                                                          0045
      AVAP=(1.-Q)*FTOT                                                  0046
      ALIQ=Q*FTOT                                                       0047
      GO TO 999                                                         0048

                                                      SUBROUTINE QFLASH
```

```
  120 CONTINUE                                                          0049
      IF(NENTH.EQ.0) GO TO 200                                          0050
C     FEED TEMPERATURE ABOVE DEW PT OR BELOW BUBBLE PT.
      NEN=1                                                             0051
      IF(NOWN.EQ.1) CALL ENERGY(TBUBPT,TDEWPT,NEN,S1,S2,S3,S4)          0052
      IF(NOWN.EQ.2) CALL ENTHAL(ALIQ,AVAP,BM,DM,BTOT,DTOT,FTOT,HD,HFEED,  0053
     1HLIQ,HVAP,NC,NF1,NP1,NP2,NVAP,QC,QF,QR,TEMP,VRTOT,X,XF,Y,YF,TBUBPT  0054
     2,TDEWPT,TFEED,NEN,S1,S2,S3,S4,NTIME,MC,MP)                        0055
  121 CONTINUE                                                          0056
      IF(TFEED.LT.TBUBPT) GO TO 122                                     0057
      IF(NTYP.EQ.1) Q=(S3-S1)/(S3-S4)                                   0058
      DO 15 I=1,NC                                                      0059
      XF(I)=0.0                                                         0060
      YF(I)=Z(I)                                                        0061
   15 CONTINUE                                                          0062
      AVAP=FTOT                                                         0063
      ALIQ=0.0                                                          0064
      GO TO 999                                                         0065
  122 CONTINUE                                                          0066
      IF(NTYP.EQ.1) Q=(S3-S2)/(S3-S4)                                   0067
      DO 16 I=1,NC                                                      0068
      XF(I)=Z(I)                                                        0069
      YF(I)=0.0                                                         0070
   16 CONTINUE                                                          0071
      AVAP=0.0                                                          0072
      ALIQ=FTOT                                                         0073
      GO TO 999                                                         0074
  200 CONTINUE                                                          0075
C     USER MUST SPECIFY FEED Q VALUE IF ENTHALPY EFFECTS ARE IGNORED.
      IF(TFEED.LT.TBUBPT) WRITE(NTP,2001) TBUBPT                        0076
 2001 FORMAT(/T2,'FEED IS BELOW ITS BUBBLE POINT OF',F8.2,1X,'DEG C',   0077
     1/T2,'ENTER FEED Q VALUE.')                                        0078
      IF(TFEED.GT.TDEWPT) WRITE(NTP,2002) TDEWPT                        0079
 2002 FORMAT(/T2,'FEED IS VAPOUR ABOVE ITS DEW POINT OF',F8.2,1X,'DEG C'  0080
     1,/T2,'ENTER FEED Q VALUE.')                                       0081
      READ(NCR,1) Q                                                     0082
      NTYP=2                                                            0083
      GO TO 121                                                         0084
```

```
  300 CONTINUE                                                          0085
C     ISENTHALPIC FLASH CALCULATION.
      NEN=1                                                             0086
      IF(NOWN.EQ.1) CALL ENERGY(TBUBPT,TDEWPT,NEN,S1,S2,S3,S4)          0087
      IF(NOWN.EQ.2) CALL ENTHAL(ALIQ,AVAP,BM,DM,BTOT,DTOT,FTOT,HD,HFEED,  0088
     1HLIQ,HVAP,NC,NF1,NP1,NP2,NVAP,QC,QF,QR,TEMP,VRTOT,X,XF,Y,YF,TBUBPT  0089
     2,TDEWPT,TFEED,NEN,S1,S2,S3,S4,NTIME,MC,MP)                        0090
      IF(S3.LT.FENTH.OR.S4.GT.FENTH) GO TO 320                          0091
      FACT=0.5                                                          0092
      FV=FACT*FTOT                                                      0093
      DELF=0.1                                                          0094
      MM=1                                                              0095
      DO 31 J=1,1000                                                    0096
      TFEED=TBUBPT+FACT*(TDEWPT-TBUBPT)                                 0097
      FV=FACT*FTOT                                                      0098
      FL=FTOT-FV                                                        0099
      IF(FV.LT.0.0.OR.FL.LT.0.0) GO TO 901                              0100
      MN=1                                                              0101
      DELT=5.                                                           0102
  310 CONTINUE                                                          0103
      NEQ=1                                                             0104
      CALL EQUCON(NEQ)                                                  0105
      SUMXF=0.                                                          0106
      DO 32 I=1,NC                                                      0107
      XF(I)=FTOT*Z(I)/(FL+FV*KF(I))                                     0108
      SUMXF=SUMXF+XF(I)                                                 0109
   32 CONTINUE                                                          0110
      IF(ABS(SUMXF-SUM1).LT.0.0001) GO TO 311                           0111
      NCALL=2                                                           0112
      CALL SEARCH(SUMXF,SUM1,TFEED,DELT,MN,NCALL)                       0113
      GO TO 310                                                         0114
  311 CONTINUE                                                          0115
      SUMYF=0.                                                          0116
      DO 33 I=1,NC                                                      0117
      YF(I)=XF(I)*KF(I)                                                 0118
      SUMYF=SUMYF+YF(I)                                                 0119
   33 CONTINUE                                                          0120
```

```
      DO 34 I=1,NC                                                          0121
      YF(I)=YF(I)/SUMYF                                                     0122
   34 CONTINUE                                                              0123
      AVAP=FV                                                               0124
      ALIQ=FL                                                               0125
      Q=FL/FTOT                                                             0126
      NEN=2                                                                 0127
      IF(NOWN.EQ.1) CALL ENERGY(TBUBPT,TDEWPT,NEN,S1,S2,S3,S4)              0128
      IF(NOWN.EQ.2) CALL ENTHAL(ALIQ,AVAP,BM,DM,BTOT,DTOT,FTOT,HD,HFEED,    0129
     1HLIQ,HVAP,NC,NF1,NP1,NP2,NVAP,QC,QF,QR,TEMP,VRTOT,X,XF,Y,YF,TBUBPT    0130
     2,TFEED,TFEED,NEN,S1,S2,S3,S4,NTIME,MC,MP)                             0131
      IF(ABS(HFEED-FENTH).LT.0.1) GO TO 999                                 0132
      NCALL=1                                                               0133
      CALL SEARCH(HFEED,FENTH,FACT,DELF,MM,NCALL)                           0134
   31 CONTINUE                                                              0135
  320 CONTINUE                                                              0136
      IF(S3.LT.FENTH) GO TO 330                                             0137
C     FEED TEMP CALC IF FEED IS SUPERCOOLED AT COLUMN PRESSURE.
      TFEED=TBUBPT-1.                                                       0138
      MN=1                                                                  0139
      DELT=1.                                                               0140
  321 CONTINUE                                                              0141
      NEN=1                                                                 0142
      IF(NOWN.EQ.1) CALL ENERGY(TBUBPT,TDEWPT,NEN,S1,S2,S3,S4)              0143
      IF(NOWN.EQ.2) CALL ENTHAL(ALIQ,AVAP,BM,DM,BTOT,DTOT,FTOT,HD,HFEED,    0144
     1HLIQ,HVAP,NC,NF1,NP1,NP2,NVAP,QC,QF,QR,TEMP,VRTOT,X,XF,Y,YF,TBUBPT    0145
     2,TDEWPT,TFEED,NEN,S1,S2,S3,S4,NTIME,MC,MP)                            0146
      IF(ABS(S2-FENTH).LT.0.1) GO TO 121                                    0147
      NCALL=1                                                               0148
      CALL SEARCH(S2,FENTH,TFEED,DELT,MN,NCALL)                             0149
      GO TO 321                                                             0150
  330 CONTINUE                                                              0151
```

```
C     FEED TEMP CALC IF FEED IS SUPERHEATED AT COLUMN PRESSURE.
      TFEED=TDEWPT+1.                                                    0152
      MN=1                                                               0153
      DELT=10.                                                           0154
  331 CONTINUE                                                           0155
      NEN=1                                                              0156
      IF(NOWN.EQ.1) CALL ENERGY(TBUBPT,TDEWPT,NEN,S1,S2,S3,S4)           0157
      IF(NOWN.EQ.2) CALL ENTHAL(ALIQ,AVAP,BM,DM,BTOT,DTOT,FTOT,HD,HFEED, 0158
     1HLIQ,HVAP,NC,NF1,NP1,NP2,NVAP,QC,QF,QR,TEMP,VRTOT,X,XF,Y,YF,TBUBPT 0159
     2,TDEWPT,TFEED,NEN,S1,S2,S3,S4,NTIME,MC,MP)                         0160
      IF(ABS(S1-FENTH).LT.0.1) GO TO 121                                 0161
      NCALL=1                                                            0162
      CALL SEARCH(S1,FENTH,TFEED,DELT,MN,NCALL)                          0163
      GO TO 331                                                          0164
C     ERROR MESSAGES ***************************************************
  901 CONTINUE                                                           0165
      WRITE(NTP,9001) FACT                                               0166
 9001 FORMAT(/T2,'ERROR IN FLASH CALCULATION.',/T2,'SEARCH FOR FEED TEMP 0167
     1ERATURE HAS FAILED.',/T2,'VAPOUR FRACTION OF FEED=',F8.4,/T2,      0168
     2'VALUE SHOULD LIE BETWEEN 0.0 AND 1.0.')                           0169
      NJ=1000                                                            0170
  999 CONTINUE                                                           0171
      WRITE(NTP,4000) Q,TFEED,TDEWPT,TBUBPT                              0172
 4000 FORMAT(/T2,'FEED Q VALUE AND TEMPERATURE DATA:',/T2,'Q:',F8.4,     0173
     115X,'FEED TEMP:',F8.2,1X,'DEG C',/T2,'DEW PT:',F8.2,1X,'DEG C',4X, 0174
     2'BUBBLE PT:',F8.2,1X,'DEG C',/)                                    0175
      RETURN                                                             0176
      END                                                                0177
```

```
      SUBROUTINE BUBCAL(IT,MMD,TBUB)                                    0001
C     BUBBLE POINT ROUTINE.
C     FOUR CALCULATION METHODS ARE AVAILABLE. CHOICE DEPENDS ON THE
C     TYPE OF EQUILIBRIUM DATA SPECIFIED.
C     CHOICE MADE BY COUNTER NEQUIL.
C     ************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ************************************************
      REAL K,LMFR,LRTOT,LSTOT,LD,LB,LPROF                               0002
      GO TO (100,200,300,400),NEQUIL                                    0003
  100 CONTINUE                                                          0004
C     CALCULATION USING ANTOINE CONSTANTS.
      TOLD=TEMP(IT)                                                     0005
      DO 11 KTIMES=1,1000                                               0006
      ASUM=PRESS(IT)                                                    0007
      BSUM=0.0                                                          0008
      DO 12 I=1,NC                                                      0009
      EFF=EFFV(I,IT)                                                    0010
      IF(MMD.GT.1) EFF=1.0                                              0011
      ANT=AN(I)-BN(I)/(CN(I)+TOLD)                                      0012
      ASUM=ASUM-X(I,IT)*EFF*10.**ANT                                    0013
      BSQ=2.303*BN(I)/((CN(I)+TOLD)**2.)                                0014
      BSUM=BSUM+X(I,IT)*EFF*BSQ*10.**ANT                                0015
   12 CONTINUE                                                          0016
      DIV=ASUM/BSUM                                                     0017
      IF(ABS(DIV).LE.0.001) GO TO 110                                   0018
      TOLD=TOLD+DIV                                                     0019
      IF(TOLD.LT.0.5) GO TO 999                                         0020
   11 CONTINUE                                                          0021
  110 CONTINUE                                                          0022
C     CHANGE IN TEMPERATURE FROM ONE ITERATION TO THE NEXT IS RESTRICTED
C     TO HALF THE CALCULATED CHANGE.
      TBUB=TOLD                                                         0023
      DIFF=TEMP(IT)-TOLD                                                0024
      IF(DIFF) 113,114,113                                              0025
  113 TEMP(IT)=TOLD+DIFF/2.                                             0026
      GO TO 999                                                         0027
  114 TEMP(IT)=TOLD                                                     0028
      GO TO 999                                                         0029
```

```
  200 CONTINUE                                                          0030
C     CALCULATION USING EQUILIBRIUM CONSTANT DATA,AS SPECIFIED BY
C     SUBROUTINE 'EQUCON' (OPTION (2) IN EQUCON).
      TOLD=TEMP(IT)*1.8+32.+459.4                                       0031
  250 CONTINUE                                                          0032
      TOLD1=TOLD/100.                                                   0033
      SUM=0.                                                            0034
      SUMD=0.                                                           0035
      DO 21 I=1,NC                                                      0036
      EFF=EFFV(I,IT)                                                    0037
      IF(MMD.GT.1) EFF=1.0                                              0038
      ALFA=EXP(AK1(I)-AK2(I)/TOLD1+AK3(I)*TOLD1)*EFF                    0039
      DA=AK3(I)+AK2(I)/(TOLD1**2.)                                      0040
      DALFA=(EXP(AK1(I)-AK2(I)/TOLD1+AK3(I)*TOLD1))*DA*EFF              0041
      SUM=SUM+(X(I,IT)*ALFA)                                            0042
      SUMD=SUMD+(X(I,IT)*DALFA)                                         0043
   21 CONTINUE                                                          0044
      FUNCT=SUM-1.                                                      0045
      FUNCTD=SUMD                                                       0046
      DIV=FUNCT/FUNCTD                                                  0047
      IF(ABS(DIV).LT.0.001) GO TO 210                                   0048
      TOLD=TOLD-DIV                                                     0049
      GO TO 250                                                         0050
  210 TNEW=(TOLD-32.-459.4)/1.8                                         0051
C     CHANGE IN TEMPERATURE FROM ONE ITERATION TO THE NEXT IS RESTRICTED
C     TO HALF THE CALCULATED CHANGE.
      TBUB=TNEW                                                         0052
      DIFF=TEMP(IT)-TNEW                                                0053
      IF(DIFF) 211,212,211                                              0054
  211 TEMP(IT)=TNEW+DIFF/2.                                             0055
      GO TO 999                                                         0056
  212 TEMP(IT)=TNEW                                                     0057
      GO TO 999                                                         0058
```

```
  300 CONTINUE                                                          0059
C     CALCULATION USING RELATIVE VOLATILITY DATA,AS SPECIFIED BY
C     SUBROUTINE 'EQUCON' (OPTION (3) IN EQUCON).
      TOLD=TEMP(IT)*1.8+32.                                             0060
  350 CONTINUE                                                          0061
      EFF=EFFV(NJ,IT)                                                   0062
      IF(MMD.GT.1) EFF=1.0                                              0063
      TOLD1=TOLD/100.                                                   0064
      AKB=(AKB1+AKB2*TOLD1+AKB3*TOLD1**2.+AKB4*TOLD1**3.)*EFF           0065
      DAKB=(AKB2+2.*AKB3*TOLD1+3.*AKB4*TOLD1**2.)*EFF                   0066
      SUM=0.                                                            0067
      SUMD=0.                                                           0068
      DO 31 I=1,NC                                                      0069
      SUM=SUM+(X(I,IT)*(AK1(I)+AK2(I)*TOLD1+AK3(I)*TOLD1**2.))          0070
      SUMD=SUMD+(X(I,IT)*(AK2(I)+2.*AK3(I)*TOLD1))                      0071
   31 CONTINUE                                                          0072
      FUNCT=AKB-(1./SUM)                                                0073
      FUNCTD=DAKB+(SUMD/(SUM**2.))                                      0074
      DIV=FUNCT/FUNCTD                                                  0075
      IF(ABS(DIV).LT.0.001) GO TO 310                                   0076
      TOLD=TOLD-DIV                                                     0077
      GO TO 350                                                         0078
  310 TNEW=(TOLD-32.)/1.8                                               0079
C     CHANGE IN TEMPERATURE FROM ONE ITERATION TO THE NEXT IS RESTRICTED
C     TO HALF THE CALCULATED CHANGE.
      TBUB=TNEW                                                         0080
      DIFF=TEMP(IT)-TNEW                                                0081
      IF(DIFF) 311,312,311                                              0082
  311 TEMP(IT)=TNEW+DIFF/2.                                             0083
      GO TO 999                                                         0084
  312 TEMP(IT)=TNEW                                                     0085
      GO TO 999                                                         0086
```

```
  400 CONTINUE                                                          0087
C     CALCULATION USING EQUILIBRIUM CONSTANT DATA, AS SPECIFIED BY
C     SUBROUTINE 'EQUCON' (OPTION (4) IN EQUCON).
      TOLD=TEMP(IT)*1.8+32.+459.4                                       0088
  450 CONTINUE                                                          0089
      T=TOLD/1000.                                                      0090
      SUM=0.                                                            0091
      SUMD=0.                                                           0092
      DO 41 I=1,NC                                                      0093
      EFF=EFFV(I,IT)                                                    0094
      IF(MMD.GT.1) EFF=1.0                                              0095
      AK=AK1(I)+AK2(I)*T+AK3(I)*T**2.+AK4(I)*T**3.                      0096
      ALFA=(AK**3.)*TOLD*EFF                                            0097
      DA=(AK**2.)*(AK1(I)+4.*AK2(I)*T+7.*AK3(I)*T**2.+10.*AK4(I)*T**3.) 0098
      DALFA=DA*EFF                                                      0099
      SUM=SUM+(X(I,IT)*ALFA)                                            0100
      SUMD=SUMD+(X(I,IT)*DALFA)                                         0101
   41 CONTINUE                                                          0102
      FUNCT=SUM-1.                                                      0103
      FUNCTD=SUMD                                                       0104
      DIV=FUNCT/FUNCTD                                                  0105
      IF(ABS(DIV).LT.0.001) GO TO 410                                   0106
      TOLD=TOLD-DIV                                                     0107
      GO TO 450                                                         0108
  410 TNEW=(TOLD-32.-459.4)/1.8                                         0109
C     CHANGE IN TEMPERATURE FROM ONE ITERATION TO THE NEXT IS RESTRICTED
C     TO HALF THE CALCULATED CHANGE.
      TBUB=TNEW                                                         0110
      DIFF=TEMP(IT)-TNEW                                                0111
      IF(DIFF) 411,412,411                                              0112
  411 TEMP(IT)=TNEW+DIFF/2.                                             0113
      GO TO 999                                                         0114
  412 TEMP(IT)=TNEW                                                     0115
  999 CONTINUE                                                          0116
      RETURN                                                            0117
      END                                                               0118
```

```
      SUBROUTINE DEWCAL(IT,MMD,TDEW)                                    0001
C     DEW POINT ROUTINE.
C     FOUR CALCULATION METHODS ARE AVAILABLE. CHOICE DEPENDS ON THE
C     TYPE OF EQUILIBRIUM DATA SPECIFIED.
C     CHOICE MADE BY COUNTER NEQUIL.
C     ************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ************************************************
      REAL K,LMFR,LRTOT,LSTOT,LD,LB,LPROF                               0002
      GO TO (100,200,300,400),NEQUIL                                    0003
  100 CONTINUE                                                          0004
C     CALCULATION USING ANTOINE CONSTANTS.
      TOLD=TEMP(IT)                                                     0005
      DO 11 KTIMES=1,1000                                               0006
      ASUM=1.                                                           0007
      BSUM=0.                                                           0008
      DO 12 I=1,NC                                                      0009
      EFF=EFFV(I,IT)                                                    0010
      IF(MMD.GT.1) EFF=1.0                                              0011
      ANT=AN(I)-BN(I)/(CN(I)+TOLD)                                      0012
      EANT=(10.**ANT)*EFF                                               0013
      ASUM=ASUM-Y(I,IT)*PRESS(IT)/EANT                                  0014
      BSUM=BSUM+Y(I,IT)*PRESS(IT)*(2.303*BN(I)/((CN(I)+TOLD)**2.))/EANT 0015
   12 CONTINUE                                                          0016
      DIV=ASUM/BSUM                                                     0017
      IF(ABS(DIV).LE.0.001) GO TO 110                                   0018
      TOLD=TOLD-DIV                                                     0019
   11 CONTINUE                                                          0020
  110 CONTINUE                                                          0021
C     CHANGE IN TEMPERATURE FROM ONE ITERATION TO THE NEXT IS RESTRICTED
C     TO HALF THE CALCULATED CHANGE.
      TDEW=TOLD                                                         0022
      DIFF=TEMP(IT)-TOLD                                                0023
      IF(DIFF) 111,112,111                                              0024
  111 TEMP(IT)=TOLD+DIFF/2.                                             0025
      GO TO 999                                                         0026
  112 TEMP(IT)=TOLD                                                     0027
      GO TO 999                                                         0028
```

```
  200 CONTINUE                                                          0029
C     CALCULATION USING EQUILIBRIUM CONSTANT DATA, AS SPECIFIED BY
C     SUBROUTINE 'EQUCON' (OPTION (2) IN EQUCON).
      TOLD=TEMP(IT)*1.8+32.+459.4                                       0030
  250 CONTINUE                                                          0031
      TOLD1=TOLD/100.                                                   0032
      SUM=0.                                                            0033
      SUMD=0.                                                           0034
      DO 21 I=1,NC                                                      0035
      EFF=EFFV(I,IT)                                                    0036
      IF(MMD.GT.1) EFF=1.0                                              0037
      ALFA=EXP(AK1(I)-AK2(I)/TOLD1+AK3(I)*TOLD1)*EFF                    0038
      DA=AK3(I)+AK2(I)/(TOLD1**2.)                                      0039
      DALFA=(EXP(AK1(I)-AK2(I)/TOLD1+AK3(I)*TOLD1))*DA*EFF              0040
      SUM=SUM+(Y(I,IT)/ALFA)                                            0041
      SUMD=SUMD+(Y(I,IT)/(ALFA**2.))*DALFA                              0042
   21 CONTINUE                                                          0043
      FUNCT=SUM-1.                                                      0044
      FUNCTD=-SUMD                                                      0045
      DIV=FUNCT/FUNCTD                                                  0046
      IF(ABS(DIV).LT.0.001) GO TO 210                                   0047
      TOLD=TOLD-DIV                                                     0048
      GO TO 250                                                         0049
  210 TNEW=(TOLD-32.-459.4)/1.8                                         0050
C     CHANGE IN TEMPERATURE FROM ONE ITERATION TO THE NEXT IS RESTRICTED
C     TO HALF THE CALCULATED CHANGE.
      TDEW=TNEW                                                         0051
      DIFF=TEMP(IT)-TNEW                                                0052
      IF(DIFF) 211,212,211                                              0053
  211 TEMP(IT)=TNEW+DIFF/2.                                             0054
      GO TO 999                                                         0055
  212 TEMP(IT)=TNEW                                                     0056
      GO TO 999                                                         0057
```

```
  300 CONTINUE                                                          0058
C     CALCULATION USING RELATIVE VOLATILITY DATA,AS SPECIFIED BY
C     SUBROUTINE 'EQUCON' (OPTION (3) IN EQUCON).
      TOLD=TEMP(IT)*1.8+32.                                             0059
  350 CONTINUE                                                          0060
      EFF=EFFV(NJ,IT)                                                   0061
      IF(MMD.GT.1) EFF=1.0                                              0062
      TOLD1=TOLD/100.                                                   0063
      AKB=(AKB1+AKB2*TOLD1+AKB3*TOLD1**2.+AKB4*TOLD1**3.)*EFF           0064
      DAKB=(AKB2+2.*AKB3*TOLD1+3.*AKB4*TOLD1**2.)*EFF                   0065
      SUM=0.                                                            0066
      SUMD=0.                                                           0067
      DO 31 I=1,NC                                                      0068
      ALFA=AK1(I)+AK2(I)*TOLD1+AK3(I)*TOLD1**2.                         0069
      DALFA=AK2(I)+2.*AK3(I)*TOLD1                                      0070
      SUM=SUM+(Y(I,IT)/ALFA)                                            0071
      SUMD=SUMD+(Y(I,IT)/(ALFA**2.))*DALFA                              0072
   31 CONTINUE                                                          0073
      FUNCT=AKB-SUM                                                     0074
      FUNCTD=DAKB+SUMD                                                  0075
      DIV=FUNCT/FUNCTD                                                  0076
      IF(ABS(DIV).LT.0.001) GO TO 310                                   0077
      TOLD=TOLD-DIV                                                     0078
      GO TO 350                                                         0079
  310 TNEW=(TOLD-32.)/1.8                                               0080
C     CHANGE IN TEMPERATURE FROM ONE ITERATION TO THE NEXT IS RESTRICTED
C     TO HALF THE CALCULATED CHANGE.
      TDEW=TNEW                                                         0081
      DIFF=TEMP(IT)-TNEW                                                0082
      IF(DIFF) 311,312,311                                              0083
  311 TEMP(IT)=TNEW+DIFF/2.                                             0084
      GO TO 999                                                         0085
  312 TEMP(IT)=TNEW                                                     0086
      GO TO 999                                                         0087
```

```
  400 CONTINUE                                                          0088
C     CALCULATION USING EQUILIBRIUM CONSTANT DATA, AS SPECIFIED BY
C     SUBROUTINE 'EQUCON' (OPTION (4) IN EQUCON).
      TOLD=TEMP(IT)*1.8+32.+459.4                                       0089
  450 CONTINUE                                                          0090
      T=TOLD/1000.                                                      0091
      SUM=0.                                                            0092
      SUMD=0.                                                           0093
      DO 41 I=1,NC                                                      0094
      EFF=EFFV(I,IT)                                                    0095
      IF(MMD.GT.1) EFF=1.0                                              0096
      AK=AK1(I)+AK2(I)*T+AK3(I)*T**2.+AK4(I)*T**3.                      0097
      ALFA=(AK**3.)*TOLD*EFF                                            0098
      DA=(AK**2.)*(AK1(I)+4.*AK2(I)*T+7.*AK3(I)*T**2.+10.*AK4(I)*T**3.) 0099
      DALFA=DA*EFF                                                      0100
      SUM=SUM+(Y(I,IT)/ALFA)                                            0101
      SUMD=SUMD+(Y(I,IT)/(ALFA**2.))*DALFA                              0102
   41 CONTINUE                                                          0103
      FUNCT=SUM-1.                                                      0104
      FUNCTD=-SUMD                                                      0105
      DIV=FUNCT/FUNCTD                                                  0106
      IF(ABS(DIV).LT.0.001) GO TO 410                                   0107
      TOLD=TOLD-DIV                                                     0108
      GO TO 450                                                         0109
  410 TNEW=(TOLD-32.-459.4)/1.8                                         0110
C     CHANGE IN TEMPERATURE FROM ONE ITERATION TO THE NEXT IS RESTRICTED
C     TO HALF THE CALCULATED CHANGE.
      TDEW=TNEW                                                         0111
      DIFF=TEMP(IT)-TNEW                                                0112
      IF(DIFF) 411,412,411                                              0113
  411 TEMP(IT)=TNEW+DIFF/2.                                             0114
      GO TO 999                                                         0115
  412 TEMP(IT)=TNEW                                                     0116
  999 CONTINUE                                                          0117
      RETURN                                                            0118
      END                                                               0119
```

```
      SUBROUTINE FLOWEQ                                                 0001
C     CALCULATION OF VAPOUR AND LIQUID FLOWRATES THROUGH THE COLUMN
C     BASED ON EQUIMOLAR OVERFLOW ASSUMPTIONS.
C     THIS SUBROUTINE IS USED FOR:
C      (1) INITIAL FLOWRATE CALCULATIONS
C      (2) FLOWRATE CALCULATIONS FOR PROBLEMS THAT IGNORE ENTHALPY EFFECTS
C          I.E. EQUIMOLAR OVERFLOW ASSUMPTIONS.
C      (3) FLOWRATE CALCULATIONS FOR THE FIRST TWO ITERATIONS IN PROBLEMS
C          CONSIDERING ENTHALPY EFFECTS, IE. NON-EQUIMOLAR OVERFLOW CASES.
C     ***************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ***************************************************
      REAL K,KF,LMFR,LRTOT,LSTOT,LD,LB,LPROF                            0002
    1 FORMAT( )                                                         0003
C     CALCULATION OF LIQUID RATES IN RECTIFYING SECTION.
  100 CONTINUE                                                          0004
      LRTOT(NP2)=RR*DTOT                                                0005
      DO 11 N=NP1,NF2,-1                                                0006
      LRTOT(N)=LRTOT(N+1)                                               0007
   11 CONTINUE                                                          0008
C     CALCULATION OF LIQUID RATES IN STRIPPING SECTION.
      LSTOT(1)=FTOT-DTOT                                                0009
      BTOT=LSTOT(1)                                                     0010
      DO 12 M=NF1,2,-1                                                  0011
      LSTOT(M)=LRTOT(NF2)+Q*FTOT                                        0012
      IF(LSTOT(M).LT.0.0) GO TO 901                                     0013
   12 CONTINUE                                                          0014
      LRTOT(NF1)=LSTOT(NF1)                                             0015
C     CALCULATION OF VAPOUR RATES IN STRIPPING SECTION.
      DO 13 M=1,NF                                                      0016
      VSTOT(M)=LSTOT(2)-BTOT                                            0017
   13 CONTINUE                                                          0018
```

```
C       CALCULATION OF VAPOUR RATES IN RECTIFYING SECTION.
        DO 14 N=NF1,NP1                                                    0019
        VRTOT(N)=VSTOT(NF)+(1.-Q)*FTOT                                     0020
     14 CONTINUE                                                           0021
        VSTOT(NF1)=VRTOT(NF1)                                              0022
        VRTOT(NP2)=0.0                                                     0023
        IF(NCOND.EQ.2) VRTOT(NP2)=DTOT                                     0024
        LSTOT(1)=BTOT                                                      0025
        GO TO 999                                                          0026
C       ERROR MESSAGES ***************************************************
    901 CONTINUE                                                           0027
        WRITE(NTP,9001) Q,LSTOT(M)                                         0028
   9001 FORMAT(/T2,'ERROR:  NEGATIVE LIQID FLOW IN STRIPPING SECTION',     0029
       1/T2,'CAUSED BY FEED CONDITION.',/T2,'Q=',F8.4,2X,'LIQUID FLOW=',   0030
       2F10.2,/T2,'ENTER AN INCREASED REFLUX RATIO TO CONTINUE RUN,OR',    0031
       3/T2,'ENTER 1000. FOR TERMINATION.')                                0032
        READ(NCR,1) RR                                                     0033
        IF(RR.LT.999.) GO TO 100                                           0034
        NJ=1000                                                            0035
    999 CONTINUE                                                           0036
        RETURN                                                             0037
        END                                                                0038
```

SUBROUTINE FLOWEQ

```
      SUBROUTINE FLOWNE                                                  0001
C     CALCULATION OF VAPOUR AND LIQUID FLOWRATES THROUGH THE COLUMN
C     BASED ON NON-EQUIMOLAR OVERFLOW.
C     THIS SUBROUTINE IS USED FOR THE THIRD AND SUBSEQUENT ITERATIONS
C     IN PROBLEMS INCLUDING ENTHALPY CONSIDERATIONS.
C     THE CHANGE IN FLOWRATE FROM ONE ITERATION TO THE NEXT IS
C     RESTRICTED AS SUGGESTED BY HOLLAND.
C       REFERENCE: HOLLAND,C.D.,'MULTICOMPONENT DISTILLATION',P 123,125.
C     **************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     **************************************************
      REAL K,KF,LMFR,LRTOT,LSTOT,LD,LB,LPROF                             0002
      IF(ICOUNT.LT.7) FLOLIM=1.2                                         0003
      IF(ICOUNT.GT.6.AND.ICOUNT.LT.12) FLOLIM=1.1                        0004
      IF(ICOUNT.GT.11) FLOLIM=1.025                                      0005
C     VAPOUR AND LIQUID FLOWS IN RECTIFYING SECTION.
      DO 11 N=NP2,NF2,-1                                                 0006
      CHEKL=LRTOT(N)                                                     0007
      CHEKLU=LRTOT(N)*FLOLIM                                             0008
      CHEKLL=LRTOT(N)/FLOLIM                                             0009
      LRTOT(N)=(DM-HVAP(N-1))*DTOT/(HVAP(N-1)-HLIQ(N))                   0010
      IF(LRTOT(N).LT.0.0) GO TO 901                                      0011
      IF(LRTOT(N).LT.CHEKLU.AND.LRTOT(N).GT.CHEKLL) GO TO 111            0012
C     THE NEW LIQUID FLOWRATE VALUE IS RESTRICTED BETWEEN LIMITS 'FLOLIM'
C     AND '1/FLOLIM' TIMES THE PREVIOUS VALUE.
      IF(LRTOT(N).GE.CHEKLU) LRTOT(N)=CHEKLU                             0013
      IF(LRTOT(N).LE.CHEKLL) LRTOT(N)=CHEKLL                             0014
      GO TO 112                                                          0015
  111 CONTINUE                                                           0016
      LRTOT(N)=LRTOT(N)-((LRTOT(N)-CHEKL)/2.)                            0017
  112 CONTINUE                                                           0018
      VRTOT(N-1)=LRTOT(N)+DTOT                                           0019
   11 CONTINUE                                                           0020

                                                        SUBROUTINE FLOWNE
```

```
C       VAPOUR AND LIQUID FLOWS IN STRIPPING SECTION.
        DO 21 M=1,NF                                                    0021
        CHEKV=VSTOT(M)                                                  0022
        CHEKVU=VSTOT(M)*FLOLIM                                          0023
        CHEKVL=VSTOT(M)/FLOLIM                                          0024
        VSTOT(M)=(HLIQ(M+1)-BM)*BTOT/(HVAP(M)-HLIQ(M+1))                0025
        IF(VSTOT(M).LT.0.0) GO TO 902                                   0026
        IF(VSTOT(M).LT.CHEKVU.AND.VSTOT(M).GT.CHEKVL) GO TO 211         0027
C       THE NEW VAPOUR FLOWRATE VALUE IS RESTRICTED BETWEEN LIMITS 'FLOLIM'
C       AND '1/FLOLIM' TIMES THE PREVIOUS VALUE.
        IF(VSTOT(M).GE.CHEKVU) VSTOT(M)=CHEKVU                          0028
        IF(VSTOT(M).LE.CHEKVL) VSTOT(M)=CHEKVL                          0029
        GO TO 212                                                       0030
  211 CONTINUE                                                          0031
        VSTOT(M)=VSTOT(M)-((VSTOT(M)-CHEKV)/2.)                         0032
  212 CONTINUE                                                          0033
        LSTOT(M+1)=VSTOT(M)+BTOT                                        0034
   21 CONTINUE                                                          0035
        LRTOT(NF1)=LSTOT(NF1)                                           0036
        VSTOT(NF1)=VRTOT(NF1)                                           0037
        VRTOT(NP2)=0.0                                                  0038
        IF(NCOND.EQ.2) VRTOT(NP2)=DTOT                                  0039
        LSTOT(1)=BTOT                                                   0040
        GO TO 999                                                       0041
```

```
C     ERROR MESSAGES ***********************************************************
  901 CONTINUE                                                                  0042
      N=N-1                                                                     0043
      WRITE(NTP,9001) N                                                         0044
 9001 FORMAT(/T2,'ERROR:  NEGATIVE LIQUID FLOW LEAVING TRAY',I4)                0045
      GO TO 903                                                                 0046
  902 CONTINUE                                                                  0047
      M=M-1                                                                     0048
      WRITE(NTP,9002) M                                                         0049
 9002 FORMAT(/T2,'ERROR:  NEGATIVE VAPOUR FLOW LEAVING TRAY',I4)                0050
  903 CONTINUE                                                                  0051
      WRITE(NTP,9003)                                                           0052
 9003 FORMAT(/T2,'POSSIBLE CAUSES OF ERROR:',/T4,'(1) REFLUX RATIO TOO L        0053
     1OW',/T4,'(2) FEED ENTHALPY TOO HIGH',/T4,'(3) ERROR IN ENTHALPY CA        0054
     2LCULATION.')                                                              0055
      WRITE(NTP,9013) QF,QC,QR                                                  0056
 9013 FORMAT(/T2,'FEED ENTHALPY=',F13.2,2X,'KJ/KGMOLE',/T2,'CONDENSER DU        0057
     1TY=',F12.2,2X,'KW',/T2,'REBOILER DUTY=',F13.2,2X,'KW')                    0058
      NJ=1000                                                                   0059
  999 CONTINUE                                                                  0060
      RETURN                                                                    0061
      END                                                                       0062
```

```
      SUBROUTINE ABSTRP                                                 0001
C     CALCULATION OF ABSORPTION AND STRIPPING FACTORS FOR ALL STAGES IN
C     THE COLUMN.
C     ABSORPTION FACTORS USED IN RECTIFYING SECTION.
C     STRIPPING FACTORS USED IN STRIPPING SECTION.
C     ***************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ***************************************************
      REAL K,LMFR,LRTOT,LSTOT,LD,LB,LPROF                               0002
      IF(NCOND.GT.1) GO TO 110                                          0003
C     ABSORPTION FACTOR FOR TOTAL-CONDENSER OPERATION.
      DO 11 I=1,NC                                                      0004
      A(I,NP2)=RR                                                       0005
   11 CONTINUE                                                          0006
      GO TO 111                                                         0007
  110 CONTINUE                                                          0008
C     ABSORPTION FACTOR FOR PARTIAL-CONDENSER OPERATION.
      DO 12 I=1,NC                                                      0009
      A(I,NP2)=LRTOT(NP2)/(K(I,NP2)*DTOT)                               0010
   12 CONTINUE                                                          0011
  111 CONTINUE                                                          0012
C     ABSORPTION FACTORS FOR STAGES FROM FEED TRAY TO TOP TRAY INCL.
      DO 13 N=NF1,NP1                                                   0013
      DO 14 I=1,NC                                                      0014
      A(I,N)=LRTOT(N)/(VRTOT(N)*K(I,N))                                 0015
   14 CONTINUE                                                          0016
   13 CONTINUE                                                          0017
C     STRIPPING FACTORS FOR STAGES FROM REBOILER TO FEED TRAY INCL.
      DO 21 M=1,NF1                                                     0018
      DO 22 I=1,NC                                                      0019
      S(I,M)=VSTOT(M)*K(I,M)/LSTOT(M)                                   0020
   22 CONTINUE                                                          0021
   21 CONTINUE                                                          0022
      RETURN                                                            0023
      END                                                               0024
```

```
      SUBROUTINE VBLB(LB,NC,NF1,S,VB,MC,MP)                              0001
C     CALCULATION OF STRIPPING SECTION.
C     THIS SUBROUTINE SOLVES THE STRIPPING SECTION MASS BALANCE-
C     EQUILIBRIUM EQUATIONS WORKING FROM THE REBOILER TOWARDS THE
C     FEED TRAY.
      DIMENSION LB(MC,MP),S(MC,MP),VB(MC,MP)                             0002
      REAL LB                                                            0003
      DO 11 I=1,NC                                                       0004
      VB(I,1)=S(I,1)                                                     0005
   11 CONTINUE                                                           0006
      DO 12 M=2,NF1                                                      0007
      DO 13 I=1,NC                                                       0008
      LB(I,M)=1.+VB(I,M-1)                                               0009
      VB(I,M)=S(I,M)*LB(I,M)                                             0010
   13 CONTINUE                                                           0011
   12 CONTINUE                                                           0012
      RETURN                                                             0013
      END                                                                0014
```

SUBROUTINE VBLB

```
      SUBROUTINE VDLD(A,LD,NC,NF2,NP1,NP2,VD,MC,MP)                    0001
C     CALCULATION OF RECTIFYING SECTION.
C     THIS SUBROUTINE SOLVES THE RECTIFYING SECTION MASS BALANCE-
C     EQUILIBRIUM EQUATIONS WORKING FROM THE CONDENSER TOWARDS THE
C     FEED TRAY.
      DIMENSION A(MC,MP),LD(MC,MP),VD(MC,MP)                           0002
      REAL LD                                                          0003
      DO 11 I=1,NC                                                     0004
      VD(I,NP1)=1.+A(I,NP2)                                            0005
   11 CONTINUE                                                         0006
      DO 12 N=NP1,NF2,-1                                               0007
      DO 13 I=1,NC                                                     0008
      LD(I,N)=A(I,N)*VD(I,N)                                           0009
      VD(I,N-1)=1.+LD(I,N)                                             0010
   13 CONTINUE                                                         0011
   12 CONTINUE                                                         0012
      RETURN                                                           0013
      END                                                              0014
```

SUBROUTINE VDLD

```
      SUBROUTINE BDCALC(A,BD,D,FTOT,LB,NC,NF1,SUMDI,VD,Z,MC,MP)          0001
C     CALCULATION OF THE B/D RATIOS AT THE FEED TRAY.
C     THIS SUBROUTINE MATCHES THE RECTIFYING AND STRIPPING SECTIONS
C     AT THE FEED TRAY.
      DIMENSION A(MC,MP),BD(MC),D(MC),LB(MC,MP),VD(MC,MP),Z(MC)          0002
      REAL LB                                                            0003
      SUMDI=0.0                                                          0004
      DO 11 I=1,NC                                                       0005
      BD(I)=A(I,NF1)*VD(I,NF1)/LB(I,NF1)                                 0006
      D(I)=FTOT*Z(I)/(1.+BD(I))                                          0007
      SUMDI=SUMDI+D(I)                                                   0008
   11 CONTINUE                                                           0009
      RETURN                                                             0010
      END                                                                0011
```

SUBROUTINE BDCALC

```
      SUBROUTINE BDCORR(B,BD,D,FTOT,NC,SLIQ,SUMD,SUMB,THETA,Z,MC,MP)          0001
C     CALCULATION OF THE CORRECTED B/D RATIOS.
C     THETA IS USED TO CORRECT THE CALCULATED B/D RATIOS. THE CORRECTED
C     VALUES ARE USED TO CALCULATE THE COMPOSITION PROFILES THROUGH
C     THE COLUMN IN THE NEXT ROUTINE.
      DIMENSION B(MC),BD(MC),D(MC),SLIQ(MP),Z(MC)                             0002
      SUMD=0.                                                                 0003
      SUMB=0.                                                                 0004
      DO 10 I=1,NC                                                            0005
      BD(I)=BD(I)*THETA                                                       0006
      D(I)=FTOT*Z(I)/(1.+BD(I))                                               0007
      B(I)=D(I)*BD(I)                                                         0008
      SUMD=SUMD+D(I)                                                          0009
      SUMB=SUMB+B(I)                                                          0010
   10 CONTINUE                                                                0011
      SLIQ(1)=SUMB                                                            0012
      RETURN                                                                  0013
      END                                                                     0014
```

SUBROUTINE BDCORR

```
      SUBROUTINE VLRECT                                                  0001
C     CALCULATION OF THE COMPOSITION PROFILES IN THE REACTIFYING
C     SECTION. THE CORRECTED VALUES OF D,THE DISTILLATE MOLAR
C     FLOWRATES,ARE USED.
C     ***************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ***************************************************
      REAL K,LMFR,LRTOT,LSTOT,LD,LB,LPROF                                0002
      SUMV=0.                                                            0003
      DO 11 I=1,NC                                                       0004
      VMFR(I,NP1)=VD(I,NP1)*D(I)                                         0005
      SUMV=SUMV+VMFR(I,NP1)                                              0006
   11 CONTINUE                                                           0007
      SVAP(NP1)=SUMV                                                     0008
      IF(NCOND.GT.1) GO TO 111                                           0009
C     TOTAL CONDENSER OPERATION.
      DO 12 I=1,NC                                                       0010
      Y(I,NP1)=VMFR(I,NP1)/SUMV                                          0011
      Y(I,NP2)=X(I,NP2)                                                  0012
   12 CONTINUE                                                           0013
      GO TO 211                                                          0014
  111 CONTINUE                                                           0015
C     PARTIAL CONDENSER OPERATION.
      SUMXK=0.                                                           0016
      DO 21 I=1,NC                                                       0017
      Y(I,NP1)=VMFR(I,NP1)/SUMV                                          0018
      Y(I,NP2)=X(I,NP2)                                                  0019
      X(I,NP2)=VRTOT(NP1)*Y(I,NP1)/(DTOT*K(I,NP2)+LRTOT(NP2))            0020
      SUMXK=SUMXK+X(I,NP2)                                               0021
   21 CONTINUE                                                           0022
      DO 22 I=1,NC                                                       0023
      X(I,NP2)=X(I,NP2)/SUMXK                                            0024
   22 CONTINUE                                                           0025
  211 CONTINUE                                                           0026

                                                      SUBROUTINE VLRECT
```

```
C       RECTIFYING SECTION.
        DO 30 N=NP1,NF2,-1                                              0027
        SUMV=0.                                                         0028
        SUML=0.                                                         0029
        DO 31 I=1,NC                                                    0030
        VMFR(I,N-1)=VD(I,N-1)*D(I)                                      0031
        LMFR(I,N)=LD(I,N)*D(I)                                          0032
        SUMV=SUMV+VMFR(I,N-1)                                           0033
        SUML=SUML+LMFR(I,N)                                             0034
   31 CONTINUE                                                          0035
        SVAP(N-1)=SUMV                                                  0036
        SLIQ(N)=SUML                                                    0037
C       MOLE FRACTION CALCULATIONS  (VAPOUR & LIQUID)
        DO 41 I=1,NC                                                    0038
        X(I,N)=LMFR(I,N)/SUML                                           0039
        Y(I,N-1)=VMFR(I,N-1)/SUMV                                       0040
   41 CONTINUE                                                          0041
   30 CONTINUE                                                          0042
        DO 50 I=1,NC                                                    0043
        LMFR(I,NP2)=D(I)*RR                                             0044
   50 CONTINUE                                                          0045
        SLIQ(NP2)=DTOT*RR                                               0046
        IF(NCOND.EQ.1) GO TO 999                                        0047
C       PARTIAL CONDENSER OPERATION.
        DO 51 I=1,NC                                                    0048
        VMFR(I,NP2)=D(I)                                                0049
   51 CONTINUE                                                          0050
        SVAP(NP2)=DTOT                                                  0051
  999 CONTINUE                                                          0052
        RETURN                                                          0053
        END                                                             0054
```

SUBROUTINE VLRECT

```
      SUBROUTINE VLSTRP                                                  0001
C     CALCULATION OF THE COMPOSITION PROFILES IN THE STRIPPING SECTION.
C     THE CORRECTED VALUES OF B,THE BOTTOMS MOLAR FLOWRATES,ARE USED.
C     ***************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ***************************************************
      REAL K,LMFR,LRTOT,LSTOT,LD,LB,LPROF                                0002
      DO 10 M=1,NF                                                       0003
      SUMV=0.                                                            0004
      SUML=0.                                                            0005
      DO 11 I=1,NC                                                       0006
      VMFR(I,M)=VB(I,M)*B(I)                                             0007
      LMFR(I,M+1)=LB(I,M+1)*B(I)                                         0008
      SUMV=SUMV+VMFR(I,M)                                                0009
      SUML=SUML+LMFR(I,M+1)                                              0010
   11 CONTINUE                                                           0011
      SVAP(M)=SUMV                                                       0012
      SLIQ(M+1)=SUML                                                     0013
C     MOLE FRACTION CALCULATIONS  (VAPOUR & LIQUID)
      DO 21 I=1,NC                                                       0014
      X(I,M+1)=LMFR(I,M+1)/SUML                                          0015
      Y(I,M)=VMFR(I,M)/SUMV                                              0016
   21 CONTINUE                                                           0017
   10 CONTINUE                                                           0018
      DO 12 I=1,NC                                                       0019
      LMFR(I,1)=B(I)                                                     0020
   12 CONTINUE                                                           0021
      RETURN                                                             0022
      END                                                                0023

                                                        SUBROUTINE VLSTRP
```

```
      SUBROUTINE THECAL                                                 0001
C     CALCULATION OF 'THETA' IN HOLLAND'S CONVERGENCE PROCEDURE.
C     THE SUBROUTINE USES A NEWTON-RAPHSON PROCEDURE TO CALCULATE
C     THETA WHICH REPRESENTS THE DEGREE OF MISMATCH OF THE RECTIFYING
C     AND STRIPPING SECTIONS AT THE FEED TRAY.
C     ***************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ***************************************************
      REAL K,LMFR,LRTOT,LSTOT,LD,LB,LPROF                               0002
      IF(THETA.LT.0.00001.OR.THETA.GT.9999.) GO TO 111                  0003
      IF(ICOUNT-1) 111,111,110                                          0004
  111 THETA=0.1                                                         0005
  110 CONTINUE                                                          0006
      SUMF=0.                                                           0007
      SUMDF=0.                                                          0008
      DO 10 I=1,NC                                                      0009
      FTHET=FTOT*Z(I)/(1.+THETA*BD(I))                                  0010
      DFTHET=FTHET*BD(I)/(1.+THETA*BD(I))                               0011
      SUMF=SUMF+FTHET                                                   0012
      SUMDF=SUMDF+DFTHET                                                0013
   10 CONTINUE                                                          0014
      FTHET=SUMF-DTOT                                                   0015
      DFTHET=-SUMDF                                                     0016
      DIV=FTHET/DFTHET                                                  0017
      THETA=THETA-DIV                                                   0018
      IF(ABS(THETA).GT.10000.) GO TO 999                                0019
      IF(THETA.GE.0.0) GO TO 112                                        0020
      THETA=THETA*(-1)                                                  0021
      GO TO 999                                                         0022
  112 CONTINUE                                                          0023
      IF(ABS(DIV).GE.0.001) GO TO 110                                   0024
  999 CONTINUE                                                          0025
      RETURN                                                            0026
      END                                                               0027
```

```
      SUBROUTINE RUNRES                                                 0001
C     ROUTINE FOR WRITING ABBREVIATED RESULTS TO THE TERMINAL.
C     ********************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ********************************************************
      DIMENSION TEMPF(MP)                                               0002
      REAL K,LMFR,LRTOT,LSTOT,LD,LB,LPROF                               0003
 1010 FORMAT(/T2,'MULTICOMPONENT DISTILLATION BY THIELE-GEDDES METHOD:.)  0004
 1011 FORMAT(/T2,'RUN NO:',2X,I3,'/',I2,'/',I2)                         0005
 1012 FORMAT(/T2,'OVERALL MATERIAL BALANCE:',/T5,'STREAM',7X,'FLOWRATE',  0006
     1 1X,'(KGMOLE/HR)',/T3,'DISTILLATE',4X,F9.4,/T3,'BOTTOMS',7X,F9.4,   0007
     2/T3,'FEED-TOTAL',4X,F9.4,/T7,'-VAPOUR',3X,F9.4,/T7,'-LIQUID',3X,    0008
     3F9.4)                                                             0009
 1013 FORMAT(/T2,,COMPONENT MOLE FRACTIONS:,,/T3,,COMP,,,2X,             0010
     1'DISTILLATE',3X,'BOTTOMS',10X,'FEED STREAM',/T33,'LIQUID',3X,     0011
     2'VAPOUR',4X,'TOTAL')                                              0012
 1014 FORMAT(T4,I2,6X,F6.4,5X,F6.4,4X,F6.4,3X,F6.4,3X,F6.4)             0013
 1015 FORMAT(/T2,'TEMPERATURE AND PRESSURE PROFILE:',/T4,'TRAY NO',10X,  0014
     1'TEMPERATURE',7X,'PRESSURE',/T18,'DEG C',7X,'DEG F',6X,'KPA',      0015
     2/T3,'CONDENSER',5X,F6.1,6X,F6.1,3X,F8.2)                          0016
 1115 FORMAT(T5,'FEED',8X,F6.1,6X,F6.1,3X,F8.2)                         0017
 1016 FORMAT(T6,I3,8X,F6.1,6X,F6.1,3X,F8.2)                             0018
 1116 FORMAT(T4,'REBOILER',5X,F6.1,6X,F6.1,3X,F8.2)                     0019
 1017 FORMAT(/T2,'NO. OF ITERATIONS=',I4,5X,'THETA=',F7.3)              0020
 1018 FORMAT(/T2,'REFLUX RATIO=',F12.2,/T2,'DISTILLATE RATE=',F9.2,2X,  0021
     1'KGMOLE/HR',/T2,'FEED Q VALUE=',F14.4,/T2,'FEED TEMPERATURE=',F8.2 0022
     2,2X,'DEG C')                                                      0023
 1019 FORMAT(/T2,'TOTAL CONDENSER OPERATION.')                          0024
 1020 FORMAT(/T2,'PARTIAL CONDENSER OPERATION.')                        0025
 1021 FORMAT(T2,'FEED ENTHALPY=',F11.2,2X,'KJ/KGMOLE',//T2,'CONDENSER DU 0026
     1TY=',F12.2,2X,'KW',/T2,'REBOILER DUTY=',F13.2,2X,'KW')            0027
 1120 FORMAT(/T2,'VAPOUR AND LIQUID FLOWRATES LEAVING TRAYS:',/T4,'TRAY  0028
     1NO',7X,'VAPOUR',7X,'LIQUID',4X,'(KGMOLE/HR)',                     0029
     2/T3,'CONDENSER',5X,F7.2,6X,F7.2)                                  0030
 1121 FORMAT(T6,I3,8X,F7.2,6X,F7.2)                                     0031
 1122 FORMAT(T5,'FEED',8X,F7.2,6X,F7.2)                                 0032
 1123 FORMAT(T4,'REBOILER',5X,F7.2,6X,F7.2)                             0033
```

```
C       TEMPERATURE PROFILE IN DEG F.
        DO 11 J=1,NP2                                                    0034
        TEMPF(J)=TEMP(J)*1.8+32.                                         0035
   11 CONTINUE                                                           0036
C       COMBINED RECTIFYING AND STRIPPING SECTION FLOW DATA.
        DO 12 N=NP2,NF1,-1                                               0037
        LPROF(N)=LRTOT(N)                                                0038
        VPROF(N)=VRTOT(N)                                                0039
   12 CONTINUE                                                           0040
        DO 13 M=NF,1,-1                                                  0041
        LPROF(M)=LSTOT(M)                                                0042
        VPROF(M)=VSTOT(M)                                                0043
   13 CONTINUE                                                           0044
C       PRESSURE PROFILE CONVERTED TO S.I. UNITS.
        FACTP2=7.50061683                                                0045
        DO 14 J=1,NP2                                                    0046
        PRES(J)=PRESS(J)/FACTP2                                          0047
   14 CONTINUE                                                           0048
C       RESULT TABLE HEADING AND RUN IDENTIFICATION NUMBER.
        WRITE(NTP,1010)                                                  0049
        WRITE(NTP,1011) NUM1,NUM2,NUM3                                   0050
C       DISTILLATE,BOTTOMS AND FEED STREAM FLOWRATES.
        WRITE(NTP,1012) DTOT,BTOT,FTOT,AVAP,ALIQ                         0051
C       STREAM COMPOSITIONS.
        WRITE(NTP,1013)                                                  0052
        WRITE(NTP,1014) (I,Y(I,NP2),X(I,1),XF(I),YF(I),Z(I),I=1,NC)      0053
```

```
C     TEMPERATURE AND PRESSURE PROFILES.
      WRITE(NTP,1015) TEMP(NP2),TEMPF(NP2),PRES(NP2)                    0054
      DO 15 J=NP1,2,-1                                                  0055
      IF(J.EQ.NF1) GO TO 115                                            0056
      MMM=J-1                                                           0057
      WRITE(NTP,1016) MMM,TEMP(J),TEMPF(J),PRES(J)                      0058
      GO TO 15                                                          0059
  115 CONTINUE                                                          0060
      WRITE(NTP,1115) TEMP(NF1),TEMPF(NF1),PRES(NF1)                    0061
   15 CONTINUE                                                          0062
      WRITE(NTP,1116) TEMP(1),TEMPF(1),PRES(1)                          0063
C     CONVERGENCE DATA.
      WRITE(NTP,1017) ICOUNT,THETA                                      0064
C     REFLUX RATIO,DISTILLATE RATE AND FEED CONDITION.
      WRITE(NTP,1018) RR,DTOT,Q,TFEED                                   0065
C     HEAT DUTIES AND CONDENSER TYPE.
      IF(NENTH.GT.0) WRITE(NTP,1021) QF,QC,QR                           0066
      IF(NCOND.EQ.1) WRITE(NTP,1019)                                    0067
      IF(NCOND.EQ.2) WRITE(NTP,1020)                                    0068
C     VAPOUR AND LIQUID FLOWRATE PROFILES.
      WRITE(NTP,1120) VPROF(NP2),LPROF(NP2)                             0069
      DO 16 J=NP1,2,-1                                                  0070
      IF(J.EQ.NF1) GO TO 116                                            0071
      MMM=J-1                                                           0072
      WRITE(NTP,1121) MMM,VPROF(J),LPROF(J)                             0073
      GO TO 16                                                          0074
  116 CONTINUE                                                          0075
      WRITE(NTP,1122) VPROF(NF1),LPROF(NF1)                             0076
   16 CONTINUE                                                          0077
      WRITE(NTP,1123) VPROF(1),LPROF(1)                                 0078
      RETURN                                                            0079
      END                                                               0080
```

```
      SUBROUTINE OUTPUT(ERROR,ERRD)                                      0001
C     ROUTINE FOR PRINTING SETS OF DETAILED RESULTS.
C     THIS ROUTINE WRITES TO A DATA-FILE WHICH IS PRINTED BY THE
C     LINE-PRINTER AT THE END OF THE RUN.
C     ************************************************
C     INSERT DIMENSION AND COMMON BLOCK 'MULTICOM' HERE.
C     ************************************************
      DIMENSION FEED(MC)                                                 0002
      REAL K,LMFR,LRTOT,LSTOT,LD,LB,LPROF                                0003
 1010 FORMAT(//T11,'MULTICOMPONENT DISTILLATION BY THIELE-GEDDES METHOD.  0004
     1',//T11,'PRINTOUT OF FULL RESULTS FROM PROGRAM THIELE-GEDDES,'      0005
     2///T11,'CONVERGENCE CRITERIA:')                                    0006
 1011 FORMAT(//T11,'THETA = ',F10.4,10X,'CONVERGENCE LIMIT=',F7.5)        0007
 1012 FORMAT(///T11,'REFLUX RATIO=',F12.4,//T11,'DISTILLATE RATE=',F9.4,  0008
     12X,'KGMOLE/HR',//T11,'DISTILLATE COMPOSITIONS:',//T15,'COMPONENT',  0009
     2T46,'MOL FRACTION')                                                0010
 1013 FORMAT(/T11,I3,3X,5A6,3X,F6.4)                                     0011
 1014 FORMAT(//T11,'NUMBER OF ITERATIONS = ',I3,T41,'THETA CONVERGENCE E  0012
     1RROR=',F10.4,/T41,'DISTILLATE FLOWRATE ERROR=',F8.4)               0013
 1015 FORMAT(/T11,'SPECIFIED AND CALCULATED DISTILLATE RATES MATCH WITHI  0014
     1N CONVERGENCE LIMIT.')                                             0015
 1110 FORMAT(///T2,110('*'),//)                                          0016
 1111 FORMAT(/T3,23('*'),/T3,'*',21X,'*',/T3,'*',1X,'RUN NO.:',2X,I3     0017
     1,'/',I2,'/',I2,1X,'*',/T3,'*',21X,'*',/T3,23('*'))                 0018
 2011 FORMAT(///T11,'TRAY NO',2X,I3,//T11,'COMPONENT NO',14X,            0019
     1'VAPOUR LEAVING',26X,'LIQUID LEAVING',/T31,'KGMOLE/HR',5X,'MOL FRA  0020
     2CTION',14X,'KGMOLE/HR',5X,'MOL FRACTION')                          0021
 2012 FORMAT(/T16,I2,12X,F10.2,5X,E12.6,13X,F10.2,5X,E12.6)              0022
 2013 FORMAT(/T31,66('-'),/T20,'TOTALS:',3X,F10.2,5X,F10.4,15X,F10.2,    0023
     15X,F10.4,/T31,66('-'),//T21,'TRAY TEMPERATURE=',F8.2,2X,'DEG C',   0024
     2//T21,'TRAY PRESSURE=',F11.2,2X,'KPA')                             0025
 2111 FORMAT(///T11,'TOTAL CONDENSER',//T11,'COMPONENT NO',13X,          0026
     1'LIQUID DISTILLATE',25X,'LIQUID REFLUX',/T31,'KGMOLE/HR',5X,'MOL F  0027
     2RACTION',14X,'KGMOLE/HR',5X,'MOL FRACTION')                        0028
 2112 FORMAT(///T11,'PARTIAL CONDENSER',//T11,'COMPONENT NO',13X,        0029
     1'VAPOUR DISTILLATE',25X,'LIQUID REFLUX',/T31,'KGMOLE/HR',5X,'MOL F  0030
     2RACTION',14X,'KGMOLE/HR',5X,'MOL FRACTION')                        0031
```

```
2113 FORMAT(/T31,66('-'),/T20,'TOTALS:',3X,F10.2,5X,F10.4,15X,F10.2,      0032
    15X,F10.4,/T31,66('-'),//T21,'CONDENSER TEMPERATURE=',F8.2,2X,       0033
    2'DEG C',//T21,'CONDENSER PRESSURE=',F11.2,2X,'KPA')                 0034
2121 FORMAT(///T2,'TRAY NO',2X,I3,4X,'* FEED TRAY *',/T2,'COMPONENT NO'  0035
    1,10X,'VAPOUR FEED',16X,'LIQUID FEED',14X,'VAPOUR LEAVING',13X,'LIQ  0036
    2UID LEAVING',/T17,'KGMOLE/HR',2X,'MOL FRACTION',4X,'KGMOLE/HR',2X,  0037
    3'MOL FRACTION',4X,'KGMOLE/HR',2X,'MOL FRACTION',4X,'KGMOLE/HR',2X,  0038
    4'MOL FRACTION')                                                     0039
2122 FORMAT(/T7,I2,7X,F10.2,2X,E12.6,3X,F10.2,2X,E12.6,3X,F10.2,2X,      0040
    1E12.6,3X,F10.2,2X,E12.6)                                            0041
2123 FORMAT(/T17,104('-'),/T5,'TOTALS:',4X,F10.2,2X,F10.4,5X,F10.2,2X,   0042
    1F10.4,5X,F10.2,2X,F10.4,5X,F10.2,2X,F10.4,/T17,104('-'),//T21,      0043
    1'TRAY TEMPERATURE=',F8.2,2X,'DEG C',//T21,'TRAY PRESSURE=',F11.2,   0044
    22X,'KPA',//T21,'FEED Q VALUE=',F12.2,//T21,'FEED TEMPERATURE=',     0045
    3F8.2,2X,'DEG C')                                                    0046
2211 FORMAT(///T11,'REBOILER',//T11,'COMPONENT NO',14X,                  0047
    1'VAPOUR LEAVING',26X,'BOTTOM PRODUCT',/T31,'KGMOLE/HR',5X,'MOL FRA  0048
    2CTION',14X,'KGMOLE/HR',5X,'MOL FRACTION')                           0049
2213 FORMAT(/T31,66('-'),/T20,'TOTALS:',3X,F10.2,5X,F10.4,15X,F10.2,     0050
    15X,F10.4,/T31,66('-'),//T21,'REBOILER TEMPERATURE=',F8.2,2X,        0051
    2'DEG C',//T21,'REBOILER PRESSURE=',F11.2,2X,'KPA')                  0052
3011 FORMAT(///T11,'OVERALL MATERIAL BALANCE:',//T11,'MOLAR PRODUCT AND  0053
    1 FEED STREAM RATES',4X,'(KGMOLE/HR)')                               0054
3012 FORMAT(//T21,'COMPONENT NO',10X,'DISTILLATE',11X,'BOTTOMS',13X,     0055
    1'FEED')                                                             0056
3013 FORMAT(/T25,I3,16X,F9.4,10X,F9.4,10X,F9.4)                          0057
3014 FORMAT(T41,50('-'),/T21,'TOTAL RATES:',11X,F9.4,10X,F9.4,10X,F9.4,  0058
    1/T41,50('-'))                                                       0059
3015 FORMAT(///T11,'HEAT DUTIES AND FEED ENTHALPY:',                     0060
    1//T11,'CONDENSER DUTY=',E16.8,2X,'KW',//T11,                        0061
    2'REBOILER DUTY=',1X,E16.8,2X,'KW',//T11,'FEED ENTHALPY=',           0062
    31X,E16.8,2X,'KJ/KGMOLE')                                            0063
3016 FORMAT(///T11,'END OF PRINTOUT.')                                   0064
```

```
C     RUN VARIABLES.
      WRITE(NLP,1110)                                                    0065
      WRITE(NLP,1111) NUM1,NUM2,NUM3                                     0066
      WRITE(NLP,1010)                                                    0067
      WRITE(NLP,1014) ICOUNT,ERROR,ERRD                                  0068
      WRITE(NLP,1011) THETA,ACCLIM                                       0069
      IF(ABS(THETA-1.0).GT.ACCLIM) WRITE(NLP,1015)                       0070
C     DISTILLATE COMPOSITION.
      WRITE(NLP,1012) RR,DTOT                                            0071
      WRITE(NLP,1013) (I,(TITLES(I,J),J=1,5),Y(I,NP2),I=1,NC)            0072
C     HEAT BALANCE.
      IF(NOWN.GT.0) WRITE(NLP,3015) QC,QR,QF                             0073
C     SPECIFICATIONS THROUGH COLUMN.
      DO 10 J=NP2,1,-1                                                   0074
      SX=0.                                                              0075
      SY=0.                                                              0076
      DO 20 I=1,NC                                                       0077
      FEED(I)=FTOT*Z(I)                                                  0078
      SX=SX+X(I,J)                                                       0079
      SY=SY+Y(I,J)                                                       0080
   20 CONTINUE                                                           0081
      IF(J.EQ.NP2) GO TO 501                                             0082
      IF(J.EQ.NF1) GO TO 510                                             0083
      IF(J.EQ.1) GO TO 502                                               0084
      MMM=J-1                                                            0085
C     DETAILS OF TRAYS OTHER THAN FEED TRAY.
      WRITE(NLP,2011) MMM                                                0086
      WRITE(NLP,2012) (I,VMFR(I,J),Y(I,J),LMFR(I,J),X(I,J),I=1,NC)       0087
      WRITE(NLP,2013) SVAP(J),SY,SLIQ(J),SX,TEMP(J),PRES(J)              0088
      GO TO 10                                                           0089
  501 CONTINUE                                                           0090
```

```
C     CONDENSER DETAILS.
      IF(NCOND.EQ.1) WRITE(NLP,2111)                                    0091
      IF(NCOND.EQ.2) WRITE(NLP,2112)                                    0092
      WRITE(NLP,2012) (I,D(I),Y(I,NP2),LMFR(I,NP2),X(I,NP2),I=1,NC)     0093
      WRITE(NLP,2113) DTOT,SY,SLIQ(NP2),SX,TEMP(NP2),PRES(NP2)          0094
      GO TO 10                                                          0095
  502 CONTINUE                                                          0096
C     REBOILER DETAILS.
      WRITE(NLP,2211)                                                   0097
      WRITE(NLP,2012) (I,VMFR(I,1),Y(I,1),LMFR(I,1),X(I,1),I=1,NC)      0098
      WRITE(NLP,2213) SVAP(1),SY,SLIQ(1),SX,TEMP(1),PRES(1)             0099
      GO TO 10                                                          0100
  510 CONTINUE                                                          0101
C     FEED TRAY DETAILS.
      WRITE(NLP,2121) NF                                                0102
      SYF=0.                                                            0103
      SXF=0.                                                            0104
      DO 51 I=1,NC                                                      0105
      YFLO=YF(I)*AVAP                                                   0106
      XFLO=XF(I)*ALIQ                                                   0107
      WRITE(NLP,2122) I,YFLO,YF(I),XFLO,XF(I),VMFR(I,NF1),Y(I,NF1),     0108
     1LMFR(I,NF1),X(I,NF1)                                              0109
      SYF=SYF+YF(I)                                                     0110
      SXF=SXF+XF(I)                                                     0111
   51 CONTINUE                                                          0112
      WRITE(NLP,2123) AVAP,SYF,ALIQ,SXF,SVAP(NF1),SY,SLIQ(NF1),SX,      0113
     1TEMP(NF1),PRES(NF1),Q,TFEED                                       0114
   10 CONTINUE                                                          0115
C     OVERALL MASS BALANCE.
      WRITE(NLP,3011)                                                   0116
      WRITE(NLP,3012)                                                   0117
      WRITE(NLP,3013) (I,D(I),B(I),FEED(I),I=1,NC)                      0118
      WRITE(NLP,3014) DTOT,BTOT,FTOT                                    0119
      WRITE(NLP,3016)                                                   0120
      WRITE(NLP,1110)                                                   0121
      RETURN                                                            0122
      END                                                               0123
```

```
C      MAINLINE PROGRAM FOR DISTILLATION-PACKAGE PLOTTING ROUTINES.
C      CONTENTS:
C           (A) MULTICOMPONENT SYSTEMS:
C                (1) TEMPERATURE PROFILE THROUGH COLUMN.
C                (2) VAPOUR AND LIQUID FLOWRATE PROFILES THROUGH COLUMN.
C                (3) DISTILLATE COMPOSITION BAR DIAGRAM.
C                (4) TRAY COMPOSITION PROFILES THROUGH COLUMN.
C
C           (B) BINARY SYSTEMS:
C                (1) BINARY BATCH MCCABE-THIELE DIAGRAM.
C                (2) BINARY CONTINUOUS MCCABE-THIELE DIAGRAM.
C                (3) BINARY CONTINUOUS PONCHON-SAVARIT DIAGRAM.
C
      DIMENSION A1(4),A2(4),A3(4),A4(4)                                  0001
      DIMENSION ALXT(4,5),ALYT(4,5)                                      0002
      DIMENSION F1(4),F2(4),F3(4),F4(4)                                  0003
      DIMENSION HEAD1(5),HEAD2(5),HEAD3(5),HEAD4(5)                      0004
      DIMENSION IBUF(1000),NCH(10),NCHARX(4),NCHARY(4),TITL(6,5)         0005
      DIMENSION X(100),Y(100),XX(100),YY(100),XAX(4,5),YAX(4,5)          0006
      DIMENSION YLEN(4)                                                  0007
      COMMON ALX1,ALY1,ALXT,ALYT,ASIZ,F1,F2,F3,F4                        0008
      COMMON NC,NCH,NCHARX,NCHARY,NP,NP2,NR,NRF,TITL                     0009
      COMMON X,Y,XX,YY,XAX,YAX,YLEN                                      0010
      DATA F1/'@ADD                 .   .   '/                           0011
      DATA F2/'@ADD                 .   .   '/                           0012
      DATA F3/'@ADD                 .   .   '/                           0013
      DATA F4/'@ADD                 .   .   '/                           0014
      DATA A1/'@ASG,AZX             .   .   '/                           0015
      DATA A2/'@ASG,AZX             .   .   '/                           0016
      DATA A3/'@ASG,AZX             .   .   '/                           0017
      DATA A4/'@ASG,AZX             .   .   '/                           0018
```

```
      DATA ASIZ,INTA,INTB,NCR,NTP/0,14,67,84,8,5/                        0019
    1 FORMAT(4(A6,1X))                                                   0020
    2 FORMAT(5A6)                                                        0021
    3 FORMAT( )                                                          0022
      WRITE(NTP,1000)                                                    0023
 1000 FORMAT(/T2,'ENTER JOB NO. AS:  NO.,DAY,MONTH (INTEGER VALUES),')   0024
      READ(NCR,3) NUM1,NUM2,NUM3                                         0025
      DNUM1=FLOAT(NUM1)                                                  0026
      DNUM2=FLOAT(NUM2)                                                  0027
      DNUM3=FLOAT(NUM3)                                                  0028
      WRITE(NTP,1010)                                                    0029
 1010 FORMAT(/T2,'ENTER TYPE OF PLOT REQUIRED. (SEE MANUAL SECTION 5)')  0030
      READ(NCR,3) NTYPE                                                  0031
      GO TO (100,200,200,300,300),NTYPE                                  0032
  100 CONTINUE                                                           0033
C     MULTICOMPONENT DISTILLATION DATA REPRESENTATION.
      WRITE(NTP,1001)                                                    0034
 1001 FORMAT(/T2,'ENTER TOTAL NUMBER OF RUNS TO BE PLOTTED.')            0035
      READ(NCR,3) NR                                                     0036
      WRITE(NTP,1002)                                                    0037
 1002 FORMAT(/T2,'ENTER NUMBER OF RUN FOR WHICH FULL PLOT IS REQUIRED.') 0038
      READ(NCR,3) NRF                                                    0039
      WRITE(NTP,1003)                                                    0040
 1003 FORMAT(/T2,'ENTER NUMBER OF TRAYS IN COLUMN,')                     0041
      READ(NCR,3) NP                                                     0042
      NP2=NP+2                                                           0043
      WRITE(NTP,1004)                                                    0044
 1004 FORMAT(/T2,'ENTER NUMBER OF COMPONENTS IN FEED.')                  0045
      READ(NCR,3) NC                                                     0046
```

```
      WRITE(NTP,1005)                                                    0047
 1005 FORMAT(/T2,,ENTER DATAFILE NAMES. (SEE MANUAL SECTION 5),)         0048
      READ(NCR,1) F1(3),F2(3),F3(3),F4(3)                                0049
      A1(3)=F1(3)                                                        0050
      A2(3)=F2(3)                                                        0051
      A3(3)=F3(3)                                                        0052
      A4(3)=F4(3)                                                        0053
      CALL ERTRAN(6,A1)                                                  0054
      CALL ERTRAN(6,A2)                                                  0055
      CALL ERTRAN(6,A3)                                                  0056
      CALL ERTRAN(6,A4)                                                  0057
C     READ AXIS TITLES AND PLOT HEADINGS FOR ALL MULTICOMPONENT PLOTS.
      CALL ERTRAN(6,'@ADD PLOTFILE.HEADING . ')                          0058
      READ(NCR,2) (HEAD1(J),J=1,5)                                       0059
      READ(NCR,2) (HEAD2(J),J=1,5)                                       0060
      READ(NCR,2) (HEAD3(J),J=1,5)                                       0061
      READ(NCR,3) NCHH1,NCHH2,NCHH3                                      0062
      CALL ERTRAN(6,'@ADD PLOTFILE.AXISTITLE . ')                        0063
      DO 11 I=1,4                                                        0064
      READ(NCR,2) (XAX(I,J),J=1,5)                                       0065
      READ(NCR,3) NCHARX(I)                                              0066
      READ(NCR,2) (YAX(I,J),J=1,5)                                       0067
      READ(NCR,3) NCHARY(I)                                              0068
   11 CONTINUE                                                           0069
      READ(NCR,3) (YLEN(I),I=1,4)                                        0070
      CALL ERTRAN(6,'@ADD PLOTFILE.TITLES . ')                           0071
      DO 12 I=1,6                                                        0072
      READ(NCR,2) (TITL(I,J),J=1,5)                                      0073
      READ(NCR,3) NCH(I)                                                 0074
   12 CONTINUE                                                           0075
      READ(NCR,3) ((ALXT(I,J),ALYT(I,J),J=1,3),I=1,4)                    0076
```

```
C       CALCOMP ROUTINE 'PLOTS' INITIATES PLOTTING PROCEDURE.
        CALL PLOTS(IBUF,1000,25)                                        0077
        CALL FACTOR(0.5)                                                0078
C       SUBROUTINE FOR COLUMN TEMPERATURE PROFILES.
        CALL PLOTTS                                                     0079
C       SUBROUTINE FOR VAPOUR AND LIQUID FLOWRATE PROFILES.
        CALL PLOTFS                                                     0080
C       SUBROUTINE FOR DISTILLATE COMPOSITION BAR DIAGRAM.
        CALL PLOTDS                                                     0081
C       SUBROUTINE FOR COMPOSITION PROFILES THROUGH THE COLUMN.
        CALL PLOTCS(ALENX)                                              0082
C       READ AND WRITE TITLES ABOVE MULTICOMPONENT PLOTS.
        ANUM1=FLOAT(NR)                                                 0083
        ANUM2=FLOAT(NRF)                                                0084
        CALL SYMBOL(1.0,7.7,ASIZ,HEAD1,0.0,NCHH1)                       0085
        CALL NUMBER(5.2,7.7,ASIZ,ANUM1,0.0,-1)                          0086
        CALL SYMBOL(1.0,7.4,ASIZ,HEAD2,0.0,NCHH2)                       0087
        CALL NUMBER(5.34,7.4,ASIZ,ANUM2,0.0,-1)                         0088
        CALL SYMBOL(1.0,7.1,ASIZ,HEAD3,0.0,NCHH3)                       0089
        ANMF=7.1                                                        0090
        ALX4=ALENX+6.0                                                  0091
        ALY4=0.0                                                        0092
        GO TO 999                                                       0093
```

MAINLINE PLOTTER

```
  200 CONTINUE                                                          0094
C     BINARY BATCH-DISTILLATION DIAGRAMS.
      CALL PLOTS(IBUF,1000,25)                                          0095
      CALL FACTOR(0.5)                                                  0096
C     SUBROUTINE FOR BINARY BATCH-DISTILLATION DIAGRAMS.
      CALL PLOTBT(ALENX,NTYPE)                                          0097
      IF(NTYPE.EQ.3) GO TO 210                                          0098
C     READ AND WRITE TITLES ON BATCH X-Y DIAGRAMS.
C     CONSTANT REFLUX OPERATION.
      CALL ERTRAN(6,,@ADD PLOTFILE.BBCTTL . ,)                          0099
      GO TO 211                                                         0100
  210 CONTINUE                                                          0101
C     VARIABLE REFLUX OPERATION.
      CALL ERTRAN(6,,@ADD PLOTFILE.BBVTTL . ,)                          0102
  211 CONTINUE                                                          0103
      READ(NCR,2) (HEAD1(I),I=1,5)                                      0104
      READ(NCR,2) (HEAD2(I),I=1,5)                                      0105
      READ(NCR,2) (HEAD3(I),I=1,5)                                      0106
      READ(NCR,2) (HEAD4(I),I=1,5)                                      0107
      READ(NCR,3) NCHB1,NCHB2,NCHB3,NCHB4                               0108
      CALL SYMBOL(1.0,11.1,ASIZ,HEAD1,0.0,NCHB1)                        0109
      CALL SYMBOL(1.0,10.8,ASIZ,HEAD2,0.0,NCHB2)                        0110
      CALL SYMBOL(1.0,10.5,ASIZ,HEAD3,0.0,NCHB3)                        0111
      CALL SYMBOL(1.0,10.2,ASIZ,HEAD4,0.0,NCHB4)                        0112
      ANMF=10.2                                                         0113
      ALX4=ALENX+6.0                                                    0114
      ALY4=0.0                                                          0115
      GO TO 999                                                         0116
```

```
  300 CONTINUE                                                          0117
C     BINARY CONTINUOUS-DISTILLATION DIAGRAMS.
      IF(NTYPE.EQ.5) WRITE(NTP,3001)                                    0118
 3001 FORMAT(/T2,'ENTER LIMITS OF ENTHALPY AXIS IN ENTH-CONC DIAGRAM.',  0119
     1/T2,'NORMALLY THESE ARE DEL PT VALUES ROUNDED CONVENIENTLY.',     0120
     2/T2,'ENTER UPPER LIMIT FIRST. (REAL MODE,FREE FORMAT)')           0121
      IF(NTYPE.EQ.5) READ(NCR,3) YST1,YST2                              0122
      CALL PLOTS(IBUF,1000,25)                                          0123
      CALL FACTOR(0.5)                                                  0124
C     SUBROUTINE FOR BINARY CONTINUOUS-DISTILLATION DIAGRAMS.
      CALL PLOTMT(ALENX,NTYPE,YST1,YST2)                                0125
      IF(NTYPE.EQ.4) GO TO 301                                          0126
C     READ AND WRITE TITLES ON PONCHON-SAVARIT ENTHALPY-CONCENTRATION DIAGRAM.
      CALL ERTRAN(6,'@ADD PLOTFILE,PSTITL . ')                          0127
      READ(NCR,2) (HEAD1(I),I=1,5)                                      0128
      READ(NCR,2) (HEAD2(I),I=1,5)                                      0129
      READ(NCR,2) (HEAD3(I),I=1,5)                                      0130
      READ(NCR,2) (HEAD4(I),I=1,5)                                      0131
      READ(NCR,3) NCHB1,NCHB2,NCHB3,NCHB4                               0132
      CALL SYMBOL(1.0,28.1,ASIZ,HEAD1,0.0,NCHB1)                        0133
      CALL SYMBOL(1.0,27.8,ASIZ,HEAD2,0.0,NCHB2)                        0134
      CALL SYMBOL(1.0,27.5,ASIZ,HEAD3,0.0,NCHB3)                        0135
      CALL SYMBOL(1.0,27.2,ASIZ,HEAD4,0.0,NCHB4)                        0136
      ANMF=27.2                                                         0137
      GO TO 302                                                         0138
```

```
  301 CONTINUE                                                          0139
C     MCCABE-THIELE X-Y DIAGRAM TITLES ADDED TO THE RUNSTREAM.
      CALL ERTRAN(6,*@ADD PLOTFILE.BCTITL . *)                          0140
  302 CONTINUE                                                          0141
C     READ AND WRITE TITLES ON MCCABE-THIELE OR PONCHON-SAVARIT X-Y DIAGRAM.
      READ(NCR,2) (HEAD1(I),I=1,5)                                      0142
      READ(NCR,2) (HEAD2(I),I=1,5)                                      0143
      READ(NCR,2) (HEAD3(I),I=1,5)                                      0144
      READ(NCR,3) NCHB1,NCHB2,NCHB3                                     0145
      CALL SYMBOL(1.0,10.8,ASIZ,HEAD1,0.0,NCHB1)                        0146
      CALL SYMBOL(1.0,10.5,ASIZ,HEAD2,0.0,NCHB2)                        0147
      CALL SYMBOL(1.0,10.2,ASIZ,HEAD3,0.0,NCHB3)                        0148
      IF(NTYPE.EQ.5) GO TO 303                                          0149
      ANMF=10.2                                                         0150
  303 CONTINUE                                                          0151
      ALX4=ALENX+6.0                                                    0152
      ALY4=0.0                                                          0153
  999 CONTINUE                                                          0154
      CALL NUMBER(2.82,ANMF,ASIZ,DNUM1,0.0,-1)                          0155
      CALL SYMBOL(3.10,ANMF,ASIZ,INTA,0.0,-1)                           0156
      CALL NUMBER(3.24,ANMF,ASIZ,DNUM2,0.0,-1)                          0157
      CALL SYMBOL(3.52,ANMF,ASIZ,INTB,0.0,-1)                           0158
      CALL NUMBER(3.66,ANMF,ASIZ,DNUM3,0.0,-1)                          0159
      CALL PLOT(ALX4,ALY4,999)                                          0160
      STOP                                                              0161
      END                                                               0162
```

MAINLINE PLOTTER

```
      SUBROUTINE PLOTTS                                                 0001
C     PLOTTING ROUTINE FOR THE TEMPERATURE PROFILE CALCULATED BY THE
C     MULTICOMPONENT DISTILLATION PROGRAM 'THIELEGEDDES'.
      DIMENSION ALXT(4,5),ALYT(4,5)                                     0002
      DIMENSION F1(4),F2(4),F3(4),F4(4)                                 0003
      DIMENSION NCH(10),NCHARX(4),NCHARY(4),TITL(6,5)                   0004
      DIMENSION XAXT(5),YAXT(5),TITLT(5)                                0005
      DIMENSION X(100),Y(100),XX(100),YY(100),XAX(4,5),YAX(4,5)         0006
      DIMENSION YLEN(4)                                                 0007
      COMMON ALX1,ALY1,ALXT,ALYT,ASIZ,F1,F2,F3,F4                       0008
      COMMON NC,NCH,NCHARX,NCHARY,NP,NP2,NR,NRF,TITL                    0009
      COMMON X,Y,XX,YY,XAX,YAX,YLEN                                     0010
    1 FORMAT( )                                                         0011
    2 FORMAT(A4)                                                        0012
    3 FORMAT(3I4)                                                       0013
    4 FORMAT(2F10.2)                                                    0014
    5 FORMAT(5A6)                                                       0015
      DATA NCHR,NCR,NPLOT,NTIMES/3,8,1,0/                               0016
      DATA ALXTR,ALXTR1,ALYTR/4.6,5.6,9.7/                              0017
      RUN1='RUN'                                                        0018
C     SET AXIS LENGTHS.
      IF(MOD(NP,2)) 102,101,102                                         0019
  101 IF(NP.LT.7) LENX=NP+1                                             0020
      IF(NP.GT.7.AND.NP.LT.17) LENX=IFIX((NP+1)/2)                      0021
      IF(NP.GT.17) LENX=8                                               0022
      GO TO 103                                                         0023
  102 IF(NP.LT.8) LENX=NP+1                                             0024
      IF(NP.GT.8.AND.NP.LT.16) LENX=(NP+1)/2                            0025
      IF(NP.GT.16) LENX=8                                               0026
  103 CONTINUE                                                          0027
      ALENX=FLOAT(LENX)                                                 0028
      ALENY=YLEN(NPLOT)                                                 0029
C     'PLOT' SETS UP ORIGIN OF AXES.
      CALL PLOT(1.0,1.0,-3)                                             0030
C     ADD DATA FILE AND READ TEMERATURE PROFILE DATA FOR FIRST RUN.
      CALL ERTRAN(6,F1)                                                 0031
```

```
      READ(NCR,2) PRD                                                    0032
      READ(NCR,3) NMAX,LFREQ,INTEQ                                       0033
      READ(NCR,4) (Y(I),X(I),I=1,NMAX)                                   0034
C     SCALE X AND Y AXES.
      CALL SCALE(X,ALENX,NMAX,1)                                         0035
      CALL SCALE(Y,ALENY,NMAX,1)                                         0036
C     SELECT AXIS TITLES.
      DO 10 J=1,5                                                        0037
      XAXT(J)=XAX(NPLOT,J)                                               0038
      YAXT(J)=YAX(NPLOT,J)                                               0039
   10 CONTINUE                                                           0040
      NCY=NCHARY(NPLOT)                                                  0041
      NCX=-NCHARX(NPLOT)                                                 0042
C     SET UP X AND Y AXES.
      CALL MAXIS(0.0,0.0,XAXT,NCX,ALENX,0.0,X(NMAX+1),X(NMAX+2),-1)      0043
      CALL AXIS(0.0,0.0,YAXT,NCY,ALENY,90.0,Y(NMAX+1),Y(NMAX+2))         0044
      NTIMES=NTIMES+1                                                    0045
C     CONSTRUCT TEMPERATURE PROFILE FOR FIRST RUN.
      CALL LINE(X,Y,NMAX,1,LFREQ,INTEQ)                                  0046
      IF(NR.EQ.1) GO TO 104                                              0047
C     CONSTRUCT TEMPERATURE PROFILES FOR REMAINING RUNS.
      NR1=NR-1                                                           0048
      DO 11 J=1,NR1                                                      0049
      READ(NCR,2) PRD                                                    0050
      READ(NCR,3) NM,LFREQ,INTEQ                                         0051
      READ(NCR,4) (YY(I),XX(I),I=1,NM)                                   0052
C     NEXT FOUR LINES SET STARTING VALUE AND INCREMENT,FOR EACH DATA
C     SET,EQUAL TO THOSE VALUES COMPUTED BY ,SCALE, FOR THE FIRST SET.
      XX(NM+1)=X(NMAX+1)                                                 0053
      XX(NM+2)=X(NMAX+2)                                                 0054
      YY(NM+1)=Y(NMAX+1)                                                 0055
      YY(NM+2)=Y(NMAX+2)                                                 0056
      NTIMES=NTIMES+1                                                    0057
      CALL LINE(XX,YY,NM,1,LFREQ,INTEQ)                                  0058
   11 CONTINUE                                                           0059
  104 CONTINUE                                                           0060
```

```
C       WRITE DIAGRAM HEADING TITLES.
        NPL=6                                                          0061
        DO 20 I=1,3                                                    0062
        ALX=ALXT(NPLOT,I)                                              0063
        ALY=ALYT(NPLOT,I)                                              0064
        NCHT=NCH(NPL)                                                  0065
        DO 21 J=1,5                                                    0066
        TITLT(J)=TITL(NPL,J)                                           0067
   21   CONTINUE                                                       0068
        NPL=NPL-1                                                      0069
        IF(NPL.LT.5) NPL=NPLOT                                         0070
        CALL SYMBOL(ALX,ALY,ASIZ,TITLT,0.0,NCHT)                       0071
   20   CONTINUE                                                       0072
C       WRITE OUT SYMBOL LEGEND TABLE ABOVE DIAGRAM.
        INTEQ=0                                                        0073
        DO 22 I=1,NTIMES                                               0074
        RUNNO=FLOAT(I)                                                 0075
        ALYTR=ALYTR-0.3                                                0076
        ALYTRI=ALYTR+0.05                                              0077
        CALL SYMBOL(ALXTR,ALYTR,ASIZ,RUNI,0.0,NCHR)                    0078
        ALXTRN=ALXTR+0.56                                              0079
        CALL NUMBER(ALXTRN,ALYTR,ASIZ,RUNNO,0.0,-1)                    0080
        INTEQ=INTEQ+2                                                  0081
        CALL SYMBOL(ALXTRI,ALYTRI,ASIZ,INTEQ,0.0,-1)                   0082
   22   CONTINUE                                                       0083
C       COMPLETE DIAGRAM BOUNDARY LINE.
        CALL PLOT(0.0,ALENY,3)                                         0084
        CALL PLOT(ALENX,ALENY,2)                                       0085
        CALL PLOT(ALENX,0.0,2)                                         0086
        EXTEND=2.                                                      0087
        IF(ALENX.LT.6.) EXTEND=4.                                      0088
        ALXI=ALENX+EXTEND                                              0089
        RETURN                                                         0090
        END                                                            0091
```

```
      SUBROUTINE PLOTFS                                                 0001
C     PLOTTING ROUTINE FOR THE VAPOUR AND LIQUID FLOWRATE PROFILES
C     CALCULATED BY THE MULTICOMPONENT DISTILLATION PROGRAM 'THIELEGEDDES',
      DIMENSION ALXT(4,5),ALYT(4,5)                                     0002
      DIMENSION F1(4),F2(4),F3(4),F4(4)                                 0003
      DIMENSION NCH(10),NCHARX(4),NCHARY(4),TITL(6,5)                   0004
      DIMENSION XAXT(5),YAXT(5),TITLT(5)                                0005
      DIMENSION X(100),Y(100),XX(100),YY(100),XAX(4,5),YAX(4,5)         0006
      DIMENSION YLEN(4)                                                 0007
      COMMON ALX1,ALY1,ALXT,ALYT,ASIZ,F1,F2,F3,F4                       0008
      COMMON NC,NCH,NCHARX,NCHARY,NP,NP2,NR,NRF,TITL                    0009
      COMMON X,Y,XX,YY,XAX,YAX,YLEN                                     0010
    1 FORMAT( )                                                         0011
    2 FORMAT(A4)                                                        0012
    3 FORMAT(3I4)                                                       0013
    4 FORMAT(2F10.2)                                                    0014
    5 FORMAT(5A6)                                                       0015
      DATA NCHR,NCR,NPLOT/5,8,2/                                        0016
      DATA ALXTF,ALXTF1,ALXTR,ALXTR1,ALYTR/4.6,5.6,4.6,4.8,9.0/         0017
C     SET AXIS LENGTHS.
      IF(MOD(NP,2)) 102,101,102                                         0018
  101 IF(NP.LT.7) LENX=NP+1                                             0019
      IF(NP.GT.7.AND.NP.LT.17) LENX=IFIX((NP+1)/2)                      0020
      IF(NP.GT.17) LENX=8                                               0021
      GO TO 103                                                         0022
  102 IF(NP.LT.8) LENX=NP+1                                             0023
      IF(NP.GT.8.AND.NP.LT.16) LENX=(NP+1)/2                            0024
      IF(NP.GT.16) LENX=8                                               0025
  103 CONTINUE                                                          0026
      ALENX=FLOAT(LENX)                                                 0027
      ALENY=YLEN(NPLOT)                                                 0028
C     'PLOT' SETS THE ORIGIN OF THE AXES.
      CALL PLOT(ALX1,0.0,-3)                                            0029
```

```
C     ADD DATA FILE AND SEARCH FOR THE CORRECT SET OF DATA. (SET 'NRF')
      CALL ERTRAN(6,F2)                                                  0030
      DO 10 J=1,NRF                                                      0031
C     LIQUID PROFILE DATA.
      READ(NCR,2) PRD                                                    0032
      READ(NCR,3) NMAX,LFREQ,INTEQ                                       0033
      READ(NCR,4) (Y(I),X(I),I=1,NMAX)                                   0034
C     VAPOUR PROFILE DATA.
      READ(NCR,2) PRD                                                    0035
      READ(NCR,3) NMAX,LFREQ1,INTEQ1                                     0036
      READ(NCR,4) (YY(I),XX(I),I=1,NMAX)                                 0037
   10 CONTINUE                                                           0038
C     SCALE AXES.
      CALL SCALE(X,ALENX,NMAX,1)                                         0039
      CALL SCALE(Y,ALENY,NMAX,1)                                         0040
C     SET VAPOUR PROFILE STARTING VALUE AND INCREMENT EQUAL TO THOSE
C     VALUES COMPUTED BY 'SCALE' FOR THE LIQUID PROFILE DATA.
      XX(NMAX+1)=X(NMAX+1)                                               0041
      XX(NMAX+2)=X(NMAX+2)                                               0042
      YY(NMAX+1)=Y(NMAX+1)                                               0043
      YY(NMAX+2)=Y(NMAX+2)                                               0044
C     SELECT AXIS TITLES.
      DO 11 J=1,5                                                        0045
      XAXT(J)=XAX(NPLOT,J)                                               0046
      YAXT(J)=YAX(NPLOT,J)                                               0047
   11 CONTINUE                                                           0048
      NCY=NCHARY(NPLOT)                                                  0049
      NCX=-NCHARX(NPLOT)                                                 0050
C     SET UP X AND Y AXES.
      CALL MAXIS(0.0,0.0,XAXT,NCX,ALENX,0.0,X(NMAX+1),X(NMAX+2),-1)      0051
      CALL AXIS(0.0,0.0,YAXT,NCY,ALENY,90.0,Y(NMAX+1),Y(NMAX+2))         0052
C     PLOT LIQUID PROFILE.
      CALL LINE(X,Y,NMAX,1,LFREQ,INTEQ)                                  0053
C     PLOT VAPOUR PROFILE.
      CALL LINE(XX,YY,NMAX,1,LFREQ1,INTEQ1)                              0054
```

```
C     SELECT DIAGRAM HEADING TITLES.
      NPL=6                                                          0055
      DO 20 I=1,3                                                    0056
      ALX=ALXT(NPLOT,I)                                              0057
      ALY=ALYT(NPLOT,I)                                              0058
      NCHT=NCH(NPL)                                                  0059
      DO 21 J=1,5                                                    0060
      TITLT(J)=TITL(NPL,J)                                           0061
   21 CONTINUE                                                       0062
      NPL=NPL-1                                                      0063
      IF(NPL.LT.5) NPL=NPLOT                                         0064
C     WRITE TITLES ABOVE DIAGRAM.
      CALL SYMBOL(ALX,ALY,ASIZ,TITLT,0.0,NCHT)                       0065
   20 CONTINUE                                                       0066
C     WRITE OUT SYMBOL LEGEND ABOVE DIAGRAM.
      ALIQ='LIQ'                                                     0067
      AVAP='VAP'                                                     0068
      NCHF=3                                                         0069
      CALL SYMBOL(ALXTF,ALYTR,ASIZ,AVAP,0.0,NCHF)                    0070
      ALXTF1=5.6                                                     0071
      CALL SYMBOL(ALXTF1,ALYTR,ASIZ,ALIQ,0.0,NCHF)                   0072
      ALYTR=ALYTR-0.3                                                0073
      ALYTR1=ALYTR+0.05                                              0074
      CALL SYMBOL(ALXTR1,ALYTR1,ASIZ,INTEQ1,0.0,-1)                  0075
      ALXTR2=ALXTR1+1.0                                              0076
      CALL SYMBOL(ALXTR2,ALYTR1,ASIZ,INTEQ,0.0,-1)                   0077
C     COMPLETE DIAGRAM BOUNDARY LINE.
      CALL PLOT(0.0,ALENY,3)                                         0078
      CALL PLOT(ALENX,ALENY,2)                                       0079
      CALL PLOT(ALENX,0.0,2)                                         0080
      ALX1=-ALX1                                                     0081
      ALY1=ALENY+4.0                                                 0082
      RETURN                                                         0083
      END                                                            0084
```

```
      SUBROUTINE PLOTDS                                                 0001
C     PLOTTING ROUTINE FOR A DISTILLATE COMPOSITION BAR DIAGRAM,CALCULATED
C     BY THE MULTICOMPONENT DISTILLATION PROGRAM 'THIELEGEDDES',
      DIMENSION ALXT(4,5),ALYT(4,5)                                     0002
      DIMENSION F1(4),F2(4),F3(4),F4(4)                                 0003
      DIMENSION NCH(10),NCHARX(4),NCHARY(4),TITL(6,5)                   0004
      DIMENSION XAXT(5),XPP(35),YAXT(5),TITLT(5)                        0005
      DIMENSION X(100),Y(100),XX(100),YY(100),XAX(4,5),YAX(4,5)         0006
      DIMENSION YLEN(4)                                                 0007
      COMMON ALX1,ALY1,ALXT,ALYT,ASIZ,F1,F2,F3,F4                       0008
      COMMON NC,NCH,NCHARX,NCHARY,NP,NP2,NR,NRF,TITL                    0009
      COMMON X,Y,XX,YY,XAX,YAX,YLEN                                     0010
    1 FORMAT( )                                                         0011
    2 FORMAT(A4)                                                        0012
    3 FORMAT(3I4)                                                       0013
    4 FORMAT(2F10.2)                                                    0014
    5 FORMAT(5A6)                                                       0015
      DATA NCR,NPLOT/8,3/                                               0016
      DATA YMAX1,YMAX2/0.0,0.2/                                         0017
C     SET AXIS LENGTHS.
      LENX=NC+1                                                         0018
      LENAX=NC-1                                                        0019
      IF(LENX.GT.18) LENX=18                                            0020
      ALENX=FLOAT(LENX)                                                 0021
      ALENAX=FLOAT(LENAX)                                               0022
      ALENY=YLEN(NPLOT)                                                 0023
      DO 10 J=1,5                                                       0024
      XAXT(J)=XAX(NPLOT,J)                                              0025
      YAXT(J)=YAX(NPLOT,J)                                              0026
   10 CONTINUE                                                          0027
      NCX=NCHARX(NPLOT)                                                 0028
      NCY=NCHARY(NPLOT)                                                 0029
C     'PLOT' SETS THE ORIGIN OF THE AXES.
      CALL PLOT(ALX1,ALY1,-3)                                           0030
```

```
C       ADD DATA FILE AND SEARCH FOR THE CORRECT DATA SET. (SET 'NRF')
        CALL ERTRAN(6,F3)                                              0031
        DO 11 J=1,NRF                                                  0032
        READ(NCR,2) PRD                                                0033
        READ(NCR,3) NMAX,LFREQ,INTEQ                                   0034
        READ(NCR,4) (X(I),Y(I),I=1,NMAX)                               0035
   11   CONTINUE                                                       0036
        Y(NMAX+1)=YMAX1                                                0037
        Y(NMAX+2)=YMAX2                                                0038
        HEIGHT=0.14                                                    0039
        XPAGE=1.0                                                      0040
        YPAGE=-0.3                                                     0041
        CALL PLOT(0.0,0.0,3)                                           0042
        CALL PLOT(ALENX,0.0,2)                                         0043
        DEL=ALENAX/(NC-1)                                              0044
C       'DO' LOOP 12 SETS OUT,AND NUMBERS,THE POSITIONS OF THE BARS
C       REPRESENTING EACH COMPONENT.
        DO 12 J=1,NC                                                   0045
        XPP(J)=XPAGE-DEL/2.                                            0046
        COMP=FLOAT(J)                                                  0047
        IF(J.GT.9) XPAGE=XPAGE-0.07                                    0048
        CALL NUMBER(XPAGE,YPAGE,HEIGHT,COMP,0.0,-1)                    0049
        IF(J.GT.9) XPAGE=XPAGE+0.07                                    0050
        XPAGE=XPAGE+DEL                                                0051
   12   CONTINUE                                                       0052
        XP=(ALENX/2.)-(NCX*HEIGHT/2.)                                  0053
        YP=-0.6                                                        0054
C       WRITE X AXIS TITLE.
        CALL SYMBOL(XP,YP,HEIGHT,XAXT,0.0,NCX)                         0055
C       SET UP Y AXIS.
        CALL AXIS(0.0,0.0,YAXT,NCY,ALENY,90.0,Y(NMAX+1),Y(NMAX+2))     0056
```

```
C       CONSTRUCT BAR DIAGRAM.
        Y1=0.0                                                          0057
        DO 13 I=1,NC                                                    0058
        Y(I)=Y(I)*ALENY                                                 0059
        CALL PLOT(XPP(I),Y1,3)                                          0060
        CALL PLOT(XPP(I),Y(I),2)                                        0061
        XPPP=XPP(I)+DEL                                                 0062
        CALL PLOT(XPPP,Y(I),2)                                          0063
        CALL PLOT(XPPP,0.0,2)                                           0064
        Y1=Y(I)                                                         0065
   13 CONTINUE                                                          0066
C       SELECT DIAGRAM HEADING TITLES.
        NPL=6                                                           0067
        DO 20 I=1,3                                                     0068
        ALX=ALXT(NPLOT,I)                                               0069
        ALY=ALYT(NPLOT,I)                                               0070
        NCHT=NCH(NPL)                                                   0071
        DO 21 J=1,5                                                     0072
        TITLT(J)=TITL(NPL,J)                                            0073
   21 CONTINUE                                                          0074
        NPL=NPL-1                                                       0075
        IF(NPL.LT.5) NPL=NPLOT                                          0076
C       WRITE TITLE ABOVE DIAGRAM.
        CALL SYMBOL(ALX,ALY,ASIZ,TITLT,0.0,NCHT)                        0077
   20 CONTINUE                                                          0078
C       COMPLETE DIAGRAM BOUNDARY LINE.
        CALL PLOT(0.0,ALENY,3)                                          0079
        CALL PLOT(ALENX,ALENY,2)                                        0080
        CALL PLOT(ALENX,0.0,2)                                          0081
        ALY1=ALENY+4.0                                                  0082
        RETURN                                                          0083
        END                                                             0084
```

```
      SUBROUTINE PLOTCS(ALENX)                                          0001
C     PLOTTING ROUTINE FOR THE LIQUID COMPOSITION PROFILE CALCULATED BY
C     THE MULTICOMPONENT DISTILLATION PROGRAM 'THIELEGEDDES'.
      DIMENSION ALXT(4,5),ALYT(4,5)                                     0002
      DIMENSION F1(4),F2(4),F3(4),F4(4)                                 0003
      DIMENSION NCH(10),NCHARX(4),NCHARY(4),TITL(6,5)                   0004
      DIMENSION XAXT(5),YAXT(5),TITLT(5)                                0005
      DIMENSION X(100),Y(100),XX(100),YY(100),XAX(4,5),YAX(4,5)         0006
      DIMENSION YLEN(4)                                                 0007
      COMMON ALX1,ALY1,ALXT,ALYT,ASIZ,F1,F2,F3,F4                       0008
      COMMON NC,NCH,NCHARX,NCHARY,NP,NP2,NR,NRF,TITL                    0009
      COMMON X,Y,XX,YY,XAX,YAX,YLEN                                     0010
    1 FORMAT( )                                                         0011
    2 FORMAT(A4)                                                        0012
    3 FORMAT(3I4)                                                       0013
    4 FORMAT(2F10.2)                                                    0014
    5 FORMAT(5A6)                                                       0015
      DATA NCHR,NCR,NPLOT,NTIMES/4,8,4,0/                               0016
      DATA ALXTR,ALXTR1,YMAX1,YMAX2/4.6,5.8,0.0,0.2/                    0017
      COMP1='COMP'                                                      0018
C     'PLOT' SETS THE ORIGIN OF THE AXES.
      CALL PLOT(0.0,ALY1,-3)                                            0019
C     ADD COMPOSITION DATA FILE.
      CALL ERTRAN(6,F4)                                                 0020
      NRFC=NRF-1                                                        0021
C     SEARCH FOR REQUIRED DATA SET.
      IF(NRFC.LT.1) GO TO 100                                           0022
      DO 10 M=1,NRFC                                                    0023
      DO 11 J=1,NC                                                      0024
      READ(NCR,2) PRD                                                   0025
      READ(NCR,3) NMAX,LFREQ,INTEQ                                      0026
      READ(NCR,4) (X(I),Y(I),I=1,NMAX)                                  0027
   11 CONTINUE                                                          0028
   10 CONTINUE                                                          0029
  100 CONTINUE                                                          0030
```

```
C     READ COMPOSITION DATA FOR LIGHT COMPONENT.
      READ(NCR,2) PRD                                                    0031
      READ(NCR,3) NMAX,LFREQ,INTEQ                                       0032
      READ(NCR,4) (X(I),Y(I),I=1,NMAX)                                   0033
      Y(NMAX+1)=YMAX1                                                    0034
      Y(NMAX+2)=YMAX2                                                    0035
C     SELECT AXIS TITLES.
      DO 12 J=1,5                                                        0036
      XAXT(J)=XAX(NPLOT,J)                                               0037
      YAXT(J)=YAX(NPLOT,J)                                               0038
   12 CONTINUE                                                           0039
      NCX=-NCHARX(NPLOT)                                                 0040
      NCY=NCHARY(NPLOT)                                                  0041
C     SET AXIS LENGTHS.
      ALENY=YLEN(NPLOT)                                                  0042
      LENX=NMAX-1                                                        0043
      ALENX=FLOAT(LENX)                                                  0044
      IF(NMAX.GT.19) ALENX=18.                                           0045
C     SCALE X AXIS. (TRAY NO.)
      CALL SCALE(X,ALENX,NMAX,1)                                         0046
C     SET UP X AND Y AXES.
      CALL MAXIS(0.0,0.0,XAXT,NCX,ALENX,0.0,X(NMAX+1),X(NMAX+2),-1)      0047
      CALL AXIS(0.0,0.0,YAXT,NCY,ALENY,90.0,Y(NMAX+1),Y(NMAX+2))         0048
      NTIMES=NTIMES+1                                                    0049
C     PLOT LIGHT COMPONENT COMPOSITION DATA.
      CALL LINE(X,Y,NMAX,1,LFREQ,INTEQ)                                  0050
C     READ AND PLOT DATA FOR REMAINING COMPONENTS.
      NC1=NC-1                                                           0051
      DO 13 J=1,NC1                                                      0052
      READ(NCR,2) PRD                                                    0053
      READ(NCR,3) NM,LFREQ,INTEQ                                         0054
      READ(NCR,4) (XX(I),YY(I),I=1,NM)                                   0055
      XX(NM+1)=X(NMAX+1)                                                 0056
      XX(NM+2)=X(NMAX+2)                                                 0057
      YY(NM+1)=Y(NMAX+1)                                                 0058
      YY(NM+2)=Y(NMAX+2)                                                 0059
      NTIMES=NTIMES+1                                                    0060
      CALL LINE(XX,YY,NM,1,LFREQ,INTEQ)                                  0061
   13 CONTINUE                                                           0062
```

```
C       DETERMINE SIZE OF SYMBOL LEGEND TABLE.
        NCNT=5                                                          0063
        ALYTRS=6.7                                                      0064
        IF(NTIMES.GT.15) NCNT=6                                         0065
        IF(NTIMES.GT.18) NCNT=7                                         0066
        IF(NTIMES.GT.21) NCNT=8                                         0067
        IF(NTIMES.GT.24) NCNT=9                                         0068
        IF(NTIMES.GT.27) NCNT=10                                        0069
        IF(NCNT.EQ.6) ALYTRS=7.0                                        0070
        IF(NCNT.EQ.7) ALYTRS=7.3                                        0071
        IF(NCNT.EQ.8) ALYTRS=7.6                                        0072
        IF(NCNT.EQ.9) ALYTRS=7.9                                        0073
        IF(NCNT.EQ.10) ALYTRS=8.2                                       0074
C       SELECT DIAGRAM HEADING TITLES.
        NPL=6                                                           0075
        DO 20 I=1,3                                                     0076
        ALX=ALXT(NPLOT,I)                                               0077
        ALY=ALYT(NPLOT,I)                                               0078
        NCHT=NCH(NPL)                                                   0079
        DO 21 J=1,5                                                     0080
        TITLT(J)=TITL(NPL,J)                                            0081
   21   CONTINUE                                                        0082
        NPL=NPL-1                                                       0083
        IF(NPL.LT.5) NPL=NPLOT                                          0084
C       WRITE TITLE ABOVE DIAGRAM.
        CALL SYMBOL(ALX,ALY,ASIZ,TITLT,0.0,NCHT)                        0085
   20   CONTINUE                                                        0086
```

```
C       WRITE OUT SYMBOL LEGEND ABOVE DIAGRAM.
        INTEQ=0                                                          0087
        NTIM=0                                                           0088
        DO 22 J=1,3                                                      0089
        ALYTR=ALYTRS                                                     0090
        DO 23 I=1,NCNT                                                   0091
        NTIM=NTIM+1                                                      0092
        COMPNO=FLOAT(NTIM)                                               0093
        ALYTR=ALYTR-0.3                                                  0094
        ALYTR1=ALYTR+0.05                                                0095
        CALL SYMBOL(ALXTR,ALYTR,ASIZ,COMP1,0.0,NCHR)                     0096
        ALXTRN=ALXTR+0.7                                                 0097
        CALL NUMBER(ALXTRN,ALYTR,ASIZ,COMPNO,0.0,-1)                     0098
        INTEQ=INTEQ+1                                                    0099
        CALL SYMBOL(ALXTR1,ALYTR1,ASIZ,INTEQ,0.0,-1)                     0100
        IF(NTIM.EQ.NTIMES) GO TO 210                                     0101
   23   CONTINUE                                                         0102
        ALXTR=ALXTR+1.5                                                  0103
        ALXTR1=ALXTR1+1.5                                                0104
   22   CONTINUE                                                         0105
  210   CONTINUE                                                         0106
C       COMPLETE DIAGRAM BOUNDARY LINE.
        CALL PLOT(0.0,ALENY,3)                                           0107
        CALL PLOT(ALENX,ALENY,2)                                         0108
        CALL PLOT(ALENX,0.0,2)                                           0109
        RETURN                                                           0110
        END                                                              0111
```

SUBROUTINE PLOTCS

```
      SUBROUTINE PLOTBT(ALENX,NTYPE)                                    0001
C     PLOTTING ROUTINE FOR CONSTRUCTION OF A MCCABE-THIELE DIAGRAM
C     FOR A BINARY BATCH-DISTILLATION SYSTEM,OPERATING UNDER EITHER
C     CONSTANT OR VARIABLE REFLUX.
      DIMENSION XAX(5),YAX(5),X(55),Y(55),XEQ(55),YEQ(55)               0002
    1 FORMAT( )                                                         0003
    2 FORMAT(I5)                                                        0004
    3 FORMAT(2(2X,F6.4))                                                0005
    4 FORMAT(F7.4,F10.2)                                                0006
    5 FORMAT(5A6)                                                       0007
      DATA ALENX,ALENY,ASIZ,NCR/10.,10.,0.14.8/                         0008
      DATA XSTRT,YSTRT,XINT,YINT/0.0,0.0,0.1,0.1/                       0009
C     ADD ELEMENT AND READ AXIS TITLES.
      CALL ERTRAN(6,'@ADD PLOTFILE,AXISMT , ')                          0010
      READ(NCR,5) (XAX(J),J=1,5)                                        0011
      READ(NCR,5) (YAX(J),J=1,5)                                        0012
      READ(NCR,1) NCAX,NCAY                                             0013
      NCX=-NCAX                                                         0014
      NCY=NCAY                                                          0015
C     'PLOT' SETS THE ORIGIN OF THE AXES.
      CALL PLOT(1.0,1.0,-3)                                             0016
C     SET UP AXES AND START CONSTRUCTION OF MCCABE-THIELE DIAGRAM.
      CALL AXIS(0.0,0.0,XAX,NCX,ALENX,0.0,XSTRT,XINT)                   0017
      CALL AXIS(0.0,0.0,YAX,NCY,ALENY,90.0,YSTRT,YINT)                  0018
      CALL PLOT(0.0,ALENY,3)                                            0019
      CALL PLOT(ALENX,ALENY,2)                                          0020
      CALL PLOT(ALENX,0.0,2)                                            0021
      CALL PLOT(0.0,0.0,3)                                              0022
      CALL PLOT(ALENX,ALENY,2)                                          0023
```

```
C     ADD DATA FILE AND READ EQUILIBRIUM DATA.
      CALL ERTRAN(6,,@ADD BDPLOT. . ')                                  0024
      READ(NCR,2) NDAT                                                  0025
      READ(NCR,3) (XEQ(I),YEQ(I),I=1,NDAT)                             0026
      XEQ(NDAT+1)=XSTRT                                                 0027
      XEQ(NDAT+2)=XINT                                                  0028
      YEQ(NDAT+1)=YSTRT                                                 0029
      YEQ(NDAT+2)=YINT                                                  0030
C     PLOT EQUILIBRIUM CURVE.
      CALL LINE(XEQ,YEQ,NDAT,1,0,1)                                     0031
C     READ AND PLOT DATA FOR COMPLETING MCCABE-THIELE DIAGRAM.
      DO 20 J=1,2                                                       0032
      READ(NCR,2) MT                                                    0033
      READ(NCR,3) (X(I),Y(I),I=1,MT)                                    0034
      READ(NCR,3) YINCPT,XD                                             0035
C     PLOT OPERATING LINE.
      YNCPT=YINCPT*10.                                                  0036
      XDA=XD*10.                                                        0037
      CALL PLOT(0.0,YNCPT,3)                                            0038
      CALL PLOT(XDA,XDA,2)                                              0039
      X(MT+1)=XSTRT                                                     0040
      X(MT+2)=XINT                                                      0041
      Y(MT+1)=YSTRT                                                     0042
      Y(MT+2)=YINT                                                      0043
C     INSERT MCCABE-THIELE STEPPED STAGES.
      CALL LINE(X,Y,MT,1,0,1)                                           0044
```

```
C       NUMBER STEPPED STAGES.
        ANO=0.                                                          0045
        MT1=MT-1                                                        0046
        DO 21 I=MT1,2,-2                                                0047
        XANO=X(I)*10.-0.1                                               0048
        YANO=Y(I)*10.+0.1                                               0049
        CALL NUMBER(XANO,YANO,ASIZ,ANO,0.0,-1)                          0050
        ANO=ANO+1.                                                      0051
   21 CONTINUE                                                          0052
   20 CONTINUE                                                          0053
        IF(NTYPE.NE.2) GO TO 999                                        0054
C       PLOT AVERAGE DISTILLATE COMPOSITION ON DIAGRAM.
        READ(NCR,3) XDAVE,XD                                            0055
        XDAV=XDAVE*10.                                                  0056
        XDAN=XDAV+0.12                                                  0057
        CALL PLOT(XDAV,0.0,3)                                           0058
        CALL PLOT(XDAV,XDAV,2)                                          0059
        YDAN=0.2                                                        0060
        HT=ASIZ                                                         0061
        ANO='XD'                                                        0062
        NCHA=2                                                          0063
        DO 30 I=1,2                                                     0064
        CALL SYMBOL(XDAN,YDAN,HT,ANO,0.0,NCHA)                          0065
        XDAN=XDAN+0.28                                                  0066
        YDAN=0.1                                                        0067
        HT=0.07                                                         0068
        ANO='AVE'                                                       0069
        NCHA=3                                                          0070
   30 CONTINUE                                                          0071
  999 CONTINUE                                                          0072
        RETURN                                                          0073
        END                                                             0074
```

SUBROUTINE PLOTBT

```
      SUBROUTINE PLOTMT(ALENX,NTYPE,YST1,YST2)                          0001
C     PLOTTING ROUTINE FOR CONSTRUCTION OF A MCCABE-THIELE OR PONCHON-
C     SAVARIT DIAGRAM FOR A BINARY CONTINUOUS-DISTILLATION SYSTEM.
      DIMENSION EXAX(5),EYAX(5),HLV(105),XYH(105)                       0002
      DIMENSION XAX(5),YAX(5),X(55),Y(55),XEQ(55),YEQ(55)               0003
      DIMENSION XOP(105),YOP(105),XQ(15),YQ(15),ZFLO(15),ZFS(15)        0004
    1 FORMAT( )                                                         0005
    2 FORMAT(I5)                                                        0006
    3 FORMAT(2(2X,F6.4))                                                0007
    4 FORMAT(F7.4,F10.2)                                                0008
    5 FORMAT(5A6)                                                       0009
    6 FORMAT(2E12.6)                                                    0010
    7 FORMAT(F6.4,E12.6)                                                0011
      DATA ASTRT,BSTRT,ASIZ,NCR/1.0,1.0,0.14.8/                         0012
      IF(NTYPE.EQ.4) GO TO 101                                          0013
C
C     CONSTRUCTION OF ENTHALPY-CONCENTRATION DIAGRAM.
C
C     ADD ELEMENT AND READ AXIS TITLES.
      CALL ERTRAN(6,'@ADD PLOTFILE.AXISPS . ')                          0014
      READ(NCR,5) (EXAX(J),J=1,5)                                       0015
      READ(NCR,5) (EYAX(J),J=1,5)                                       0016
      READ(NCR,1) NCEAX,NCEAY                                           0017
      NCEX=-NCEAX                                                       0018
      NCEY=NCEAY                                                        0019
      ALENX=10.                                                         0020
      ALENY=15.                                                         0021
```

```
C     'PLOT' SETS THE ORIGIN OF THE ENTHALPY DIAGRAM.
C     (IE. LOWER LEFT-HAND CORNER)
      CALL PLOT(1.0,13.0,-3)                                          0022
      ASTRT=0.0                                                       0023
      BSTRT=-12.0                                                     0024
      XSTRT=0.0                                                       0025
      XINT=0.1                                                        0026
      YSTRT=YST2                                                      0027
      YINT=(YST1-YST2)/ALENY                                          0028
C     'AXIS' SETS UP THE Y AXIS.
      CALL AXIS(0.0,0.0,EYAX,NCEY,ALENY,90.0,YSTRT,YINT)              0029
      XST=ABS(YST2/YINT)                                              0030
C     'AXIS' SETS UP THE X AXIS.
      CALL AXIS(0.0,XST,EXAX,NCEX,ALENX,0.0,XSTRT,XINT)               0031
C     ADD DATA FILE.
      CALL ERTRAN(6,'@ADD BDPLOT. . ')                                0032
C     READ COORDINATES OF KEY POINTS ON ENTHALPY DIAGRAM.
C     (IE. 'DEL' POINTS AND FEED,BOTTOMS AND DISTILLATE POSITIONS)
      READ(NCR,6) DELD,DELB                                           0033
      READ(NCR,7) XD,HD                                               0034
      READ(NCR,7) XB,HB                                               0035
      READ(NCR,7) Z,HF                                                0036
      XD=XD*10.                                                       0037
      XB=XB*10.                                                       0038
      Z=Z*10.                                                         0039
      DELD=DELD/YINT+XST                                              0040
      DELB=DELB/YINT+XST                                              0041
      HD=HD/YINT+XST                                                  0042
      HB=HB/YINT+XST                                                  0043
      HF=HF/YINT+XST                                                  0044
C     INSERT LINES REPRESENTING OVERALL BALANCES ON ENTHALPY DIAGRAM.
      CALL PLOT(XB,HB,3)                                              0045
      CALL PLOT(XB,DELB,2)                                            0046
      CALL PLOT(XD,DELD,2)                                            0047
      CALL PLOT(XD,HD,2)                                              0048
      CALL SYMBOL(Z,HF,ASIZ,1,0.0,-1)                                 0049
```

```
      DELDD=DELD+0.2                                                    0050
      XDD=XD+ASIZ                                                       0051
      DELXD=DELD+0.1                                                    0052
      DELBB=DELB-0.2                                                    0053
      XBB=XB+ASIZ                                                       0054
      DELXB=DELB-0.3                                                    0055
      HFF=HF+0.2                                                        0056
      ZFF=Z-0.2                                                         0057
C     MARK KEY POINTS ON ENTHALPY DIAGRAM.
      CALL SYMBOL(ZFF,HFF,ASIZ,35,0.0,-1)                               0058
      CALL SYMBOL(XD,DELDD,ASIZ,28,0.0,-1)                              0059
      CALL SYMBOL(XDD,DELXD,ASIZ,33,0.0,-1)                             0060
      CALL SYMBOL(XB,DELBB,ASIZ,28,0.0,-1)                              0061
      CALL SYMBOL(XBB,DELXB,ASIZ,31,0.0,-1)                             0062
C     READ NUMBER OF DATA POINTS AVAILABLE FOR PLOTTING ENTHALPY CURVES.
      READ(NCR,2) NENTH                                                 0063
C     'DO' LOOP 11 READS DATA AND PLOTS THE ENTHALPY CURVES.
C     LIQUID ENTHALPY DATA SUPPLIED FIRST IN DATA FILE BY 'BINARY'.
      DO 11 I=1,2                                                       0064
      READ(NCR,7) (XYH(J),HLV(J),J=1,NENTH)                             0065
      XYH(NENTH+1)=XSTRT                                                0066
      XYH(NENTH+2)=XINT                                                 0067
      HLV(NENTH+1)=YSTRT                                                0068
      HLV(NENTH+2)=YINT                                                 0069
C     PLOT ENTHALPY CURVES.
      CALL LINE(XYH,HLV,NENTH,1,0,1)                                    0070
   11 CONTINUE                                                          0071
C     COMPLETE BOUNDARY LINE.
      CALL PLOT(0.0,ALENY,3)                                            0072
      CALL PLOT(ALENX,ALENY,2)                                          0073
      CALL PLOT(ALENX,0.0,2)                                            0074
      CALL PLOT(0.0,0.0,2)                                              0075
  101 CONTINUE                                                          0076

                                                      SUBROUTINE PLOTMT
```

```
C
C       CONSTRUCTION OF X-Y DIAGRAM.
C
        ALENX=10.                                                       0077
        ALENY=10.                                                       0078
C       ADD ELEMENT AND READ AXIS TITLES.
        CALL ERTRAN(6,'@ADD PLOTFILE.AXISMT . ')                        0079
        READ(NCR,5) (XAX(J),J=1,5)                                      0080
        READ(NCR,5) (YAX(J),J=1,5)                                      0081
        READ(NCR,1) NCAX,NCAY                                           0082
        NCX=-NCAX                                                       0083
        NCY=NCAY                                                        0084
C       'PLOT' SETS THE ORIGIN OF THE AXES.
        CALL PLOT(ASTRT,BSTRT,-3)                                       0085
        XSTRT=0.0                                                       0086
        XINT=0.1                                                        0087
        YSTRT=0.0                                                       0088
        YINT=0.1                                                        0089
C       SET UP AXES AND START CONSTRUCTION OF X-Y DIAGRAM.
        CALL AXIS(0.0,0.0,XAX,NCX,ALENX,0.0,XSTRT,XINT)                 0090
        CALL AXIS(0.0,0.0,YAX,NCY,ALENY,90.0,YSTRT,YINT)                0091
        CALL PLOT(0.0,ALENY,3)                                          0092
        CALL PLOT(ALENX,ALENY,2)                                        0093
        CALL PLOT(ALENX,0.0,2)                                          0094
        CALL PLOT(0.0,0.0,3)                                            0095
        CALL PLOT(ALENX,ALENY,2)                                        0096
        IF(NTYPE.EQ.5) GO TO 102                                        0097
C       ADD DATA FILE IF ENTHALPY-CONC. DIAGRAM HAS NOT BEEN PLOTTED.
        CALL ERTRAN(6,'@ADD BDPLOT. . ')                                0098
  102 CONTINUE                                                          0099
C       READ EQUILIBRIUM CURVE DATA.
        READ(NCR,2) NDAT                                                0100
        READ(NCR,3) (XEQ(I),YEQ(I),I=1,NDAT)                           0101
        XEQ(NDAT+1)=XSTRT                                               0102
        XEQ(NDAT+2)=XINT                                                0103
        YEQ(NDAT+1)=YSTRT                                               0104
        YEQ(NDAT+2)=YINT                                                0105
```

```
C       PLOT EQUILIBRIUM CURVE.
        CALL LINE(XEQ,YEQ,NDAT,1,0,1)                                   0106
        IF(NTYPE.EQ.5) GO TO 103                                        0107
C       READ OPERATING LINE DATA FOR MCCABE-THIELE X-Y DIAGRAM.
        READ(NCR,2) NQDAT                                               0108
        READ(NCR,3) (XQ(I),YQ(I),I=1,NQDAT)                            0109
        DO 10 I=1,NQDAT                                                 0110
        XQ(I)=XQ(I)*10.                                                 0111
        YQ(I)=YQ(I)*10.                                                 0112
   10 CONTINUE                                                          0113
C       PLOT OPERATING LINES ON MCCABE-THIELE X-Y DIAGRAM.
        YB=YQ(1)                                                        0114
        XB=XQ(1)                                                        0115
        CALL PLOT(XB,YB,3)                                              0116
        DO 20 I=2,NQDAT                                                 0117
        CALL PLOT(XQ(I),YQ(I),2)                                        0118
   20 CONTINUE                                                          0119
        GO TO 104                                                       0120
  103 CONTINUE                                                          0121
C       READ OPERATING LINE DATA FOR PONCHON-SAVARIT X-Y DIAGRAM.
        DO 12 I=1,2                                                     0122
        READ(NCR,2) NPTS                                                0123
        READ(NCR,3) (XOP(J),YOP(J),J=1,NPTS)                           0124
        XOP(NPTS+1)=XSTRT                                               0125
        XOP(NPTS+2)=XINT                                                0126
        YOP(NPTS+1)=YSTRT                                               0127
        YOP(NPTS+2)=YINT                                                0128
C       PLOT OPERATING LINES ON PONCHON-SAVARIT X-Y DIAGRAM.
        CALL LINE(XOP,YOP,NPTS,1,0,1)                                   0129
   12 CONTINUE                                                          0130
  104 CONTINUE                                                          0131
C       READ DATA FOR CONSTRUCTING STEPPED STAGES ON X-Y DIAGRAM.
        READ(NCR,2) MT                                                  0132
        READ(NCR,3) (X(I),Y(I),I=1,MT)                                  0133
        READ(NCR,2) NN                                                  0134
        READ(NCR,4) (ZFS(I),ZFLO(I),I=2,NN)                             0135
        IF(NTYPE.EQ.4) GO TO 105                                        0136
```

```
C     READ COORDINATES OF POINT OF INTERSECTION OF OPERATING LINES
C     FOR PONCHON-SAVARIT X-Y DIAGRAM.
      READ(NCR,3) XQ(2),YQ(2)                                          0137
      XQ(2)=XQ(2)*10.                                                  0138
      YQ(2)=YQ(2)*10.                                                  0139
  105 CONTINUE                                                         0140
      X(MT+1)=XSTRT                                                    0141
      X(MT+2)=XINT                                                     0142
      Y(MT+1)=YSTRT                                                    0143
      Y(MT+2)=YINT                                                     0144
C     INSERT MCCABE-THIELE STEPPED STAGES.
      CALL LINE(X,Y,MT,1,0,1)                                          0145
C     NUMBER STEPPED STAGES.
      ANO=0.                                                           0146
      MT1=MT-1                                                         0147
      DO 21 I=2,MT1,2                                                  0148
      XANO=X(I)*10.-0.1                                                0149
      YANO=Y(I)*10.+0.1                                                0150
      CALL NUMBER(XANO,YANO,ASIZ,ANO,0.0,-1)                           0151
      ANO=ANO+1.                                                       0152
   21 CONTINUE                                                         0153
      YP1=0.21                                                         0154
      YP2=0.0                                                          0155
      INT=22                                                           0156
C     INSERT FEED AND SIDE STREAM Q LINES.
      DO 30 I=2,NN                                                     0157
      Z=ZFS(I)*10.                                                     0158
      Z1=Z+0.05                                                        0159
      IF(ZFLO(I).GT.0.0) NZ=35                                         0160
      IF(ZFLO(I).LT.0.0) NZ=48                                         0161
      CALL SYMBOL(Z1,YP1,ASIZ,NZ,0.0,-1)                               0162
      CALL PLOT(Z,0.0,3)                                               0163
      CALL PLOT(Z,Z,2)                                                 0164
      CALL PLOT(XQ(I),YQ(I),2)                                         0165
   30 CONTINUE                                                         0166
      RETURN                                                           0167
      END                                                              0168
```